

Model 102-A
GENERAL PURPOSE
COMPUTER

OPERATION & MAINTENANCE
MANUAL VOLUME I

The National Cash Register Company
Dayton, Ohio

model
102-A
GENERAL
PURPOSE
COMPUTER

**operation
and
maintenance
manual
volume 1**

THE NATIONAL CASH REGISTER COMPANY
DAYTON, OHIO

Copyright 1956
© The National Cash Register Company
Dayton, Ohio

FOREWORD

This instruction manual is provided as a basic source of technical information on the Model 102-A General Purpose Computer manufactured by The National Cash Register Company, Dayton, Ohio. This manual also includes a description of the supplementary input-output equipment which can be used in this computing system. For convenience, the manual is contained in two separate volumes.

Volume I contains descriptive data, installation procedures, explanations of the theory of operation, and operating maintenance instructions. Volume II contains the schematics, logical networks and equations, and the block explanations. The information provided in these two volumes is sufficiently detailed to enable qualified personnel to operate and completely service the computer.

Supplementary input-output equipment obtained from vendors other than this company, are modified, as required, for operation in this computing system. These modifications are made by this company and are described in detail along with the operation and maintenance of this equipment as part of the 102-A system. Appropriate publications are referenced for more complete information on these units.

Table of Contents

SECTION I - DESCRIPTION

GENERAL INFORMATION	I-1
Basic System	I-1
Optional Input-Output Equipment	I-2
Model 126 Magnetic Tape Handling Unit	I-2
IBM Summary Punches	I-3
PHYSICAL CHARACTERISTICS	I-3
Model 102-A Computing System	I-3
Computer	I-3
Operator's Console	I-16

SECTION II - INSTALLATION

GENERAL INFORMATION	II-1
Preliminary Requirements	II-1
Filing and Storage	II-1
Thermal	II-2
System Layout	II-2
System's Records	II-4

SECTION III - FUNCTIONAL SPECIFICATIONS

INTRODUCTION	III-1
Basic Presentation of Data	III-1
Word Structure	III-1
Configuration of Octal Numbers	III-1
Configuration of Decimal Numbers	III-2
Computer Program	III-2
Configuration of Commands	III-2
MEMORY	III-3
Main Memory	III-3
Recirculating Registers	III-3
Word Channel	III-4
Special Addresses	III-4
Channel Selection	III-5

Access to the Memory	III-5
Access Time	III-5
Timing Channels	III-5
Command Code	III-7
Arithmetic Commands	III-7
Logical and Transfer Commands	III-10
Input-Output Equipment Commands	III-11

SECTION IV - THEORY

INTRODUCTION	IV-1
POWER SUPPLIES	IV-1
Clamp and Bias Voltages	IV-2
Filament Bias for Regulator Tubes	IV-4
Comparator Circuit	IV-4
Relay Sequencing	IV-4
LOGICAL CIRCUITS	IV-6
Flip-Flops, Drivers, Cathode-Followers, One-Shots	IV-7
Clock Products and Delays	IV-10
Examples of Logical Design	IV-11
MEMORY ELECTRONICS	IV-18
Drum and Heads	IV-18
Functional Description of Memory Elements	IV-19
Main Memory	IV-19
Recirculating Registers	IV-19
Clock Channels	IV-22
Word Channel	IV-22
Synchronizing Pulse	IV-23
Magnetic Techniques	IV-23
Electronic Techniques	IV-24
Recirculating Registers	IV-24
Main Memory Electronics	IV-27
Clock Circuit	IV-33
LOGICAL STRUCTURE	IV-35
Flow Diagram	IV-35
INPUT-OUTPUT ROUTINES AND CIRCUITS	IV-40
Flexowriter Circuitry and Operation	IV-40
Input to Computer	IV-40
Output to Flexowriter	IV-44
Computer - IBM Circuitry and Operation	IV-51
General Information	IV-51

Preparing Cards in Hopper	IV-52
IBM Control Panel	IV-52
Reading Procedure in the Computer	IV-52
Punching Procedure in the Computer.	IV-56
Computer-Tape Unit Circuitry and Operation	IV-57
General Information.	IV-57
Block Search Routine	IV-58
Read Tape Routine	IV-58
Write Tape Routine	IV-59

SECTION V - OPERATION

INTRODUCTION	V-1
Controls (Operators)	V-1
Computer Control Panel	V-1
Flexowriter	V-3
Test Panel	V-6
Controls (Automatic)	V-7
Alarm Checks	V-7
FILLING THE COMPUTER	V-8
Flexowriter	V-8
Manual Filling	V-8
Automatic Filling	V-11
Model 126 Magnetic Tape Unit (readout)	V-11
IBM Equipment (readout)	V-11
READ OUT FROM THE COMPUTER	V-13
Flexowriter	V-13
Model 126 Magnetic Tape Unit (Record)	V-15
IBM Equipment (Punching Cards)	V-15
SYSTEM OPERATING PROCEDURE	V-16
Start (System ON).	V-16
Filling Operation	V-16
Computation	V-16
Read Out Operation	V-16
Shutdown (System OFF)	V-16
FUNDAMENTALS OF PROGRAMMING	V-17
General Information	V-17
Techniques of Programming	V-28

SECTION VI - MAINTENANCE

INTRODUCTION	VI-1
Maintenance Program	VI-1
Precautions	VI-1
Maintenance Equipment	VI-2
Spare Parts	VI-2
Test Equipment	VI-2
Preventive Maintenance Schedule	VI-3
Preventive Maintenance Check List	VI-3
OPERATIONAL TESTING	VI-4
Reliability Program	VI-4
Part I	VI-4
Part II	VI-8
Part III	VI-9
Procedure for Filling Reliability Routine from Punched Paper Tape	VI-10
Procedure for Writing Reliability Routine on Magnetic Tape	VI-10
Procedure for Filling Reliability Routine from Magnetic Tape	VI-12
Command Test Program	VI-12
Subroutine A	VI-14
Subroutine C	VI-14
Procedure for Filling the Command Test Program	VI-15
Marginal Checking	VI-15
MEASUREMENT AND ADJUSTMENT	VI-16
Power Supplies	VI-16
Voltages	VI-16
Ripple and Adjustment Range	VI-24
Memory	VI-24
Head Installation and Radial Adjustment	VI-24
Tangential Adjustment	VI-26
Clock Channel	VI-26
Word Channel (M_w)	VI-28
Recirculating Registers	VI-30
Main Memory	VI-31
FLEXOWRITER MAINTENANCE	VI-35
MISCELLANEOUS	VI-36
Relays	VI-36
Air Filters	VI-36
Controls and Indicators	VI-37
Lubrication and Motor Inspection	VI-38
Mechanical Inspection	VI-38

COMPONENT TESTING	VI-38
Plug-ins	VI-38
5687 and 5963 Drivers, Logical	VI-39
5687 and 5963 Flip-Flop Drivers	VI-39
12AT7 One-Shot, Unclamped	VI-40
5687 One-Shot, Clamped	VI-41
12AT7 and 12BH7 Flip-Flops, Clamped	VI-41
12AT7 Flip-Flop, Unclamped	VI-42
5963 Delay Line and Cathode Follower	VI-43
5965 Special Dual Cathode Follower, Logical	VI-43
12AT7 Dual Cathode Follower	VI-44
5881 Cathode Follower Driver	VI-44
6BQ7 - A Dual Relay Driver	VI-44
6BQ7 - A Clipper	VI-45
5687 Dual Phase Inverter	VI-45
12AT7 Schmitt Trigger	VI-46
Quintuple Input Coupler	VI-46
Tubes	VI-47
Diodes	VI-47
AUXILIARY EQUIPMENT	VI-49
Model 126 Magnetic Tape Unit	VI-49
IBM Equipment	VI-51
General	VI-51
Operational Testing	VI-51

APPENDIX

GLOSSARY	A-1
BASIC SYSTEMS AND ARITHMETIC PROCESSES	B-1
Binary Number System	B-1
Octal Number System	B-4
Conversions Between Number Systems	B-6
BASIC SYMBOLIC LOGIC	C-1
List of Identities in Boolean Algebra	C-2
LOGICAL CIRCUIT DEVELOPMENT	D-1
Logical Sum, Product, and Negation in Electronic Form	D-1
PROGRAMS	E-1
Reliability	E-1
Command Test	E-12
Tape Unit Read-Write	E-14
Consecutive Block Search and Address Check	E-20
IBM Program	E-22

Fill Test Pattern	E-28
Erase Main Memory	E-28
Compare All Channels with Channel 0	E-28
Minimum Access Test	E-29
Memory Sum	E-29
Shifted Memory Sum	E-30
COMPONENT INSTALLATION AND LOCATION CHARTS	F-1
Plug-In Unit Cross-Reference List	F-1
Plug-In Unit Locations by Chassis	F-2
Plug-In Unit Complement and Locations	F-7
Tube Complement and Locations	F-9
Relay Conversion Chart	F-10
Fuse Complement (location, size, and function)	F-11
Switches and Adjustments	F-12
SPARE PARTS	G-1
FLIP-FLOP AND MEMORY OUTPUT PROPOSITION CHECK POINTS	H-1
DRAWING LIST (Breakdown)	I-1

Tables

SECTION I

I -1.	Dimensions	I-5
I -2.	Power Requirements	I-5
I -3.	Principal D-C Voltages	I-13

SECTION II

II -1.	Thermal Characteristics	II-2
II -2.	Interconnection and Power Cabling	II-4

SECTION III

III-1.	Octal Number Sign Code	III-1
III-2.	Decimal Number Sign Code	III-2
III-3.	Binary Code for Filling and Printing Decimal Digits	III-2
III-4.	Command List	III-6

SECTION IV

IV-1.	Principal D-C Voltages	IV-3
IV-2.	Flip-Flop and Logical Drivers	IV-9
IV-3.	Truth Table for Binary Counter	IV-12
IV-4.	Truth Table for Pulse Counter	IV-13
IV-5.	Truth Table for Octal Digit Counter	IV-15
IV-6.	Truth Table for Serial Adder	IV-16
IV-7.	Truth Table for Extract Circuit	IV-17
IV-8.	Bit Pattern for Channel Selection	IV-29
IV-9.	Flexowriter-to-Computer Input Settings	IV-41
IV-10.	Information Flow Between Computer and Flexowriter	IV-47
IV-11.	Computer-to-Flexowriter Conversion Chart	IV-48
IV-12.	Reading a Row of an IBM Card	IV-55
IV-13.	Arrangement of Bits in Tape Address Word	IV-58

SECTION V

V -1.	Decimal to Binary Conversion Code	V-5
V -2.	Octal to Binary Conversion Code	V-5
V -3.	Register Contents as Card is Read	V-12
V -4.	Program Format	V-20

V-5.	Program Format	V-22
V-6.	Program Format	V-25
V-7.	Program Format	V-27
V-8.	Conversion of an Integral Decimal Number to an Equivalent Binary Number.	V-29
V-9.	Conversion of a Fractional Decimal Number to an Equivalent Binary Number.	V-33
V-10.	Conversion of a Fractional Binary Number to a Fractional Decimal Number	V-37

SECTION VI

VI-1.	Error Printouts for Part I	VI-11
VI-2.	Error Printouts for Parts II and III	VI-11
VI-3.	Voltages Used in Marginal Checking	VI-15
VI-4.	Marginal Range for Various Modes of Operation.	VI-16
VI-5.	Filament Voltage Checkpoints	VI-17
VI-6.	Adjustable Voltages	VI-25
VI-7.	Alignment Procedures for Recirculating Register	VI-33
VI-8.	Main Memory Gain and D-C Level Controls	VI-35
VI-9.	Relation between Flexowriter Positions and the Fill and Print Relays	VI-36
VI-10.	Diode Discard Specifications	VI-47
VI-11.	Acceptance Specifications for Old Type Diodes	VI-48
VI-12.	Acceptance Specifications for New Type Diodes	VI-49

APPENDIX B

B-I.	Binary Addition	B-1
B-II.	Binary Multiplication	B-1
B-III.	Binary, Octal, and Decimal Equivalents	B-4
B-IV.	Octal Addition	B-5
B-V.	Octal Multiplication	B-5

APPENDIX C

C-I.	Truth Table for Negation	C-1
C-II.	Truth Table for Logical Sum	C-1
C-III.	Truth Table for Logical Product	C-1
C-IV.	Identity Truth Table	C-2

APPENDIX D

D-I.	Sum and Product Truth Tables Expressed in Voltages	D-3
------	---	-----

Table of Illustrations

Frontispiece:	Model 102-A Computer System	xvi
Figure I-1.	Typical Model 102-A System	I-4
Figure I-2.	Model 102-A Computer (exploded view)	I-7
Figure I-3.	Model 102-A General Purpose Computer-- (front view).	I-8
Figure I-4.	Model 102-A General Purpose Computer-- (rear view).	I-9
Figure I-5.	Model 102-A General Purpose Computer-- (right side).	I-10
Figure I-6.	Model 102-A General Purpose Computer-- (left side)	I-11
Figure II-1.	Typical System Layout	II-3
Figure III-1.	Memory Channel and Sector Designations	III-3
Figure III-2.	Buffer Register	III-4
Figure IV-1.	Voltage Regulation in the 225-V Supply	IV-2
Figure IV-2.	Relay Sequencing	IV-5
Figure IV-3.	Logical Structure of the Computer	IV-7
Figure IV-4.	Typical Unclamped Flip-Flop	IV-8
Figure IV-5.	Idealized Waveforms for an Unclamped Flip-Flop	IV-8
Figure IV-6.	Typical Clamped One-Shot and Idealized Waveforms.	IV-11
Figure IV-7.	Idealized Waveforms of Grid Propositions	IV-11
Figure IV-8.	Predicated Waveforms	IV-13
Figure IV-9.	Idealized Waveforms of Pulse Counter.	IV-14
Figure IV-10.	Idealized Waveforms of Digit Counter	IV-14
Figure IV-11.	Logical Circuit for Serial Adder	IV-17
Figure IV-12.	Logical Circuitry for Extract Operation	IV-18
Figure IV-13.	Logical Circuitry for Extract Operation-- (simplified).	IV-18
Figure IV-14.	Memory Drum Assembly.	IV-18
Figure IV-15.	One-Word Recirculating Register	IV-19
Figure IV-16.	Diagram of E Register	IV-20
Figure IV-17.	Inserting Information into the E Register	IV-20
Figure IV-18.	Buffer Register	IV-21
Figure IV-19.	Word Channel on the Drum	IV-23
Figure IV-20.	Locating an Address on the Drum	IV-23
Figure IV-21.	Magnetic Head.	IV-24
Figure IV-22.	Typical Record Pattern	IV-24
Figure IV-23.	Pulse Patterns Induced in a Read Head	IV-24
Figure IV-24.	Simplified Schematic of 5687 Record Tube	IV-25

Figure IV-25.	Typical Waveforms in the Playback Circuit . . .	IV-26
Figure IV-26.	Clipper Output Waveforms	IV-26
Figure IV-27.	Idealized Waveforms in the Record/Playback Cycle of the E Recirculating Register	IV-27
Figure IV-28.	Idealized Waveforms in the E Flip-Flop Register	IV-27
Figure IV-29.	Simplified Schematic of a Main Memory Pre- amplifier	IV-30
Figure IV-30.	Idealized Waveforms of Record and Playback Process in the Main Memory	IV-32
Figure IV-31.	Idealized Clock Waveforms	IV-33
Figure IV-32.	Idealized Waveforms of Clipper Outputs and Sampling Clock	IV-34
Figure IV-33.	Logical Structure of the Computer	IV-35
Figure IV-34.	Operation of a Fill Relay in the Computer.	IV-42
Figure IV-35.	Idealized Waveforms in the Start Circuit	IV-45
Figure IV-36.	IBM Control Panel	IV-53
Figure IV-37.	Block Diagram of Computer Tape Information Transfer	IV-59
Figure V-1.	Computer Control Panel	V-2
Figure V-2.	Flexowriter	V-4
Figure V-3.	Flexowriter Character Punched on Paper Tape	V-6
Figure V-4.	Computer Test Panel	V-7
Figure VI-1.	Upper Power Supply No. 3 (internal view)	VI-18
Figure VI-2.	Upper Power Supply No. 3 (external view)	VI-18
Figure VI-3.	Upper Power Supply No. 4 (internal view)	VI-19
Figure VI-4.	Upper Power Supply No. 4 (external view)	VI-19
Figure VI-5.	Lower Power Supply No. 2 (internal view)	VI-20
Figure VI-6.	Lower Power Supply No. 4 (internal view)	VI-21
Figure VI-7.	Lower Power Supply No. 3 (external view)	VI-21
Figure VI-8.	Fan Contactor Panel	VI-23
Figure VI-9.	Head Adjustment Set and Headmount	VI-27
Figure VI-10.	Head Adjustment Set in Use	VI-28
Figure VI-11.	Output Waveforms for Recirculating Registers	VI-31
Figure VI-12.	E Register Waveforms	VI-32
Figure VI-13.	Contactor Panel	VI-36
Figure VI-14.	Upper Tube Chassis No. 2 (internal view)	VI-37



Model 102-A System

Description

GENERAL INFORMATION

The Model 102-A General Purpose Computer is a high-speed, internally programmed, fully electronic, digital computing device. It has been designed to accurately perform complicated mathematical functions on an infinite variety of problems, and for extremely flexible usage with the best commercially available input-output equipment.

Specifically, the computer can operate and control up to seven Model 126 Magnetic Tape Handling Units and two IBM punch card machines. Thus, by simply varying the combination and/or number of input-output devices which can be used with the computer, the most efficient adaptation of the system to a great many applications may be accomplished, providing an almost unlimited extension of the computer's versatility.

Other features of design provide ease of operation and maintenance, convenient size and mobility, large capacity and fast access memory storage, decimal printing and filling, automatic filling from punched tape and magnetic tape, punched card input and output, and manual Flexowriter input and automatic output.

BASIC SYSTEM

The basic 102-A system includes the computer, operator's console, and an electric typewriter (Flexowriter)*. The operator's console contains the necessary switches and indicators for starting, stopping, and operating the computer. The Flexowriter serves as the

basic input-output device for the computer.

The Model 102-A is a general purpose, automatic, digital computer. "General Purpose" assures sufficient machine intelligence to solve any problem that has been satisfactorily formulated in numerical terms. It is "digital" in the sense that the data processed by the machine is a digital representation of quantities rather than physical. "Automatic" signifies that after the start of computation the computer is capable of proper execution of all prestored instructions without the aid of an operator. Thus, reliable automatic high-speed arithmetic calculation, the power to make elementary decisions, and ability to modify internally stored commands or to perform operations upon data per a set of stored commands equip this machine for a wide range of applications from general business problems to complex scientific analysis.

The Model 102-A General Purpose Computer will perform twenty-five different arithmetic, logical, and in-out commands, with a mean total operating time of sixty milliseconds for arithmetic and logical commands. The computer's memory provides a permanent storage capacity of 1024 words (see Word in the Glossary); an eight-word buffer register for input-output devices and for a quick-access memory during computation; four one-word recirculating registers to hold information being operated on by the computer; and basic timing pulses to synchronize all operations

*Manufactured by Commercial Controls Corporation, Rochester 2, New York.

in the computer.

Data may be entered into the computer manually from the Flexowriter, semi-automatically from perforated paper tape, and automatically from punched cards or from computer - controlled Model 126 Magnetic Tape Handling Units. The computer will print output data automatically on its Flexowriter in either octal or decimal form, will perforate paper tape in either octal or decimal form for reproducing hard copy or for re-entering the data into the computer at some later time, will feed data directly into IBM punch card machines, or will record data by means of a Model 126 Magnetic Tape Handling Unit for later use.

The computer control panel and Flexowriter are both mounted on the operator's console. All the controls and indicators necessary to operate and fill the computer are contained on the panel and the Flexowriter.

The Flexowriter is an electric typewriter which is modified for use as the basic input-output device for the Model 102-A computer. Attached to the Flexowriter are a paper tape punch and paper tape reader. The Flexowriter connects to the computer through the operator's console, but has its own power cable and must be plugged into the power line separately. To feed information into the computer, the Flexowriter may be operated manually by a typist or automatically by means of the paper tape reader. This device reads six-hole perforated tape at the rate of about seven characters per second. Tapes may be prepared in advance on any standard Flexowriter; they may be punched while information is being entered into the computer manually, or duplicated by reading one tape and simultaneously punching another. In all cases a typed record is also made, so that information being entered into the computer

and/or punched on tape may be checked. These tapes provide a useful storage medium for standard lists of instructions.

When serving as an output printer for the computer, the Flexowriter will print alphabetic as well as numeric data, print in two colors, and use both upper-case (capital) and lower-case characters.

Also, the output from the computer can be punched on paper tape. When not actively involved in computer filling or printing, the Flexowriter may be used in its conventional manner. (See "Flexowriter Operating Instruction Manual," published by Commercial Controls Corp. Rochester 2, New York.)

OPTIONAL INPUT-OUTPUT EQUIPMENT

Model 126 Magnetic Tape Handling Unit

This is an auxiliary high-capacity memory storage device, suitable, without design modification, for use with the Model 102-A General Purpose Computer. It is manufactured by The National Cash Register Company, Electronics Division.

This machine performs three primary functions in response to operational commands received by it from the computer which it serves. In response to the command RECORD it will record, in magnetic form on its tape, information received from the computer. In response to the command SEARCH, it will search for and locate any desired portion of the tape. In response to the command READ, it will read-out, in the form of electrical pulses, information stored magnetically on the tape. Maximum access time required in any search operation is 160 seconds; average access time is 80 seconds.

The Model 126 contains up to 1200 feet

of magnetic tape, providing storage space for over 130,000 "words" of information. Since the Model 102-A computer is capable of operating with as many as seven tape handling units, it is possible to expand the computer's memory storage capacity by this means to nearly a million "words". For complete information on this unit, see "The Model 126 Magnetic Tape Handling Unit Operation and Maintenance Manual", published by The National Cash Register Company, Electronics Division, Hawthorne, California.

IBM Summary Punches

The Model 102-A computer is capable of accepting data from, and transmitting data to, IBM cards. Two modified IBM machines (Types 523, 514, or equivalent) are required to provide this feature: one to read, and one to punch the cards. The modifications, made by IBM, are electrical in nature and may be switched in or out as desired so that the machines may be used in their normal fashion with other IBM equipment.

The IBM machines are connected by multiple-wire Summary Punch Cables to the 102-A. One end of a cable is permanently attached to the Summary Punch and the other end is equipped with a multiple-contact connector unit which is plugged into the computer cable connector panel. The IBM machines have their own power cables and these must be plugged into the power line separately.

In reading data from cards into the computer, 56 or fewer columns of data or four complete instructions may be read from each card. Results associated with data read from a deck of IBM cards may be, after computation, punched on the same cards by having the deck moved from the reader to the punch while the results are stored in the computer.

The following publications provide more complete operating instructions for

these units:

1. Gang Summary Punch, Type 523, PRINCIPLES OF OPERATION AUTOMATIC SUMMARY PUNCHES; Form 52-3403-2, Published by International Business Machines Corp.: New York 22, New York.

2. Reproducing Punch, Type 514, OPERATOR'S GUIDE, Form 22-8485-1, Published by International Business Machines Corp.; New York 22, New York

PHYSICAL CHARACTERISTICS

MODEL 102-A COMPUTING SYSTEM

The principle physical and electrical specifications covering the Model 102-A computer and optional input-output equipment are given in Tables I-1 and I-2. Shown in Figure I-1 is one system arrangement consisting of the computer, operator's console and Flexowriter, and one Model 126 Magnetic Tape Handling Unit. As has been pointed out, many variations of the system are possible.

Computer

The Model 102-A computer cabinet is fabricated of aluminum and mounted on four Darnell Steel casters, providing mobility for relocation of the unit, as required. The cabinet contains the electronic computing and control circuits with their associated power supplies, and testing and metering facilities. All four sides are removable for maintenance purposes. A hinged panel near the top of one end of the cabinet permits vision of and access to the computer test panel. All cables enter and leave the computer cabinet at the bottom center of one side. Air circulation through the cabinet is expedited by a self-contained blower which exhausts directly upward through the shutters at the top center of the cabinet. Two air intakes with fiberglass filters are located on the bottom of the computer cabinet. The major components comprising the computer are located and identified on Figure I-2 and are de-

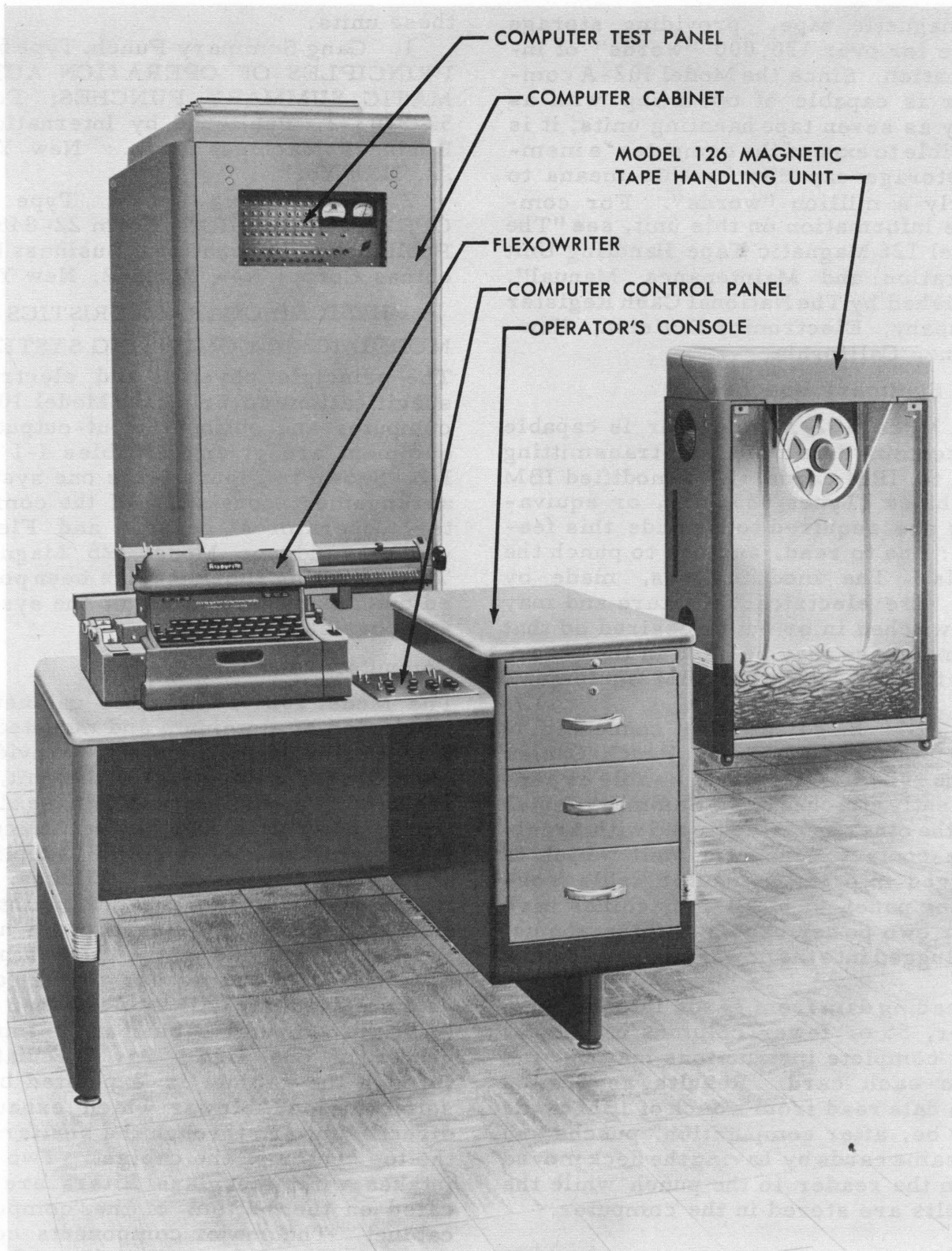


Figure I-1. Typical Model 102-A System

Unit	D"	L"	H"	Wt. (lb)
Model 102-A Computer	30	59	73*	1700
Operator's Console	33	46	35**	200
Flexowriter	21	30	11	75
Model 126 Magnetic Tape Handling Unit	30	18	48	400
IBM 523 Punch	26	41	52	705
* With casters.				
** With Flexowriter.				

TABLE I-1. Dimensions

Unit	Voltage	Power	Recommended Line and Circuit Breaker
Model 102-A Computer	230 + 5%, 57-63* 7.7KVA, 71% cycles, single phase power factor		50 amp
Operator's Console			
Flexowriter	115	.23KVA	10-30 amp
Model 126 Magnetic Tape Handling Unit	115	1.1 KVA	20-30 amp
IBM 523 Punch	115	.88KVA	15-30 amp
* Modification for 208-V, 60 cycle, single phase operation optional at no extra charge.			

TABLE I-2. Power Requirements

scribed in the following paragraphs. Figures I-3 through 6 present the computer, with side panels off, from all four sides.

Power Supply: The power required to start and run the Model 102-A computer, as shown in TABLE I-2, is about 7.7kva of 230 volts, single phase, sixty cycle, alternating current. This input power is converted into the various d-c voltages used in the computer. As noted in the above Table, suitable modification will be made if the available power is

208-v, 60 cycle, alternating current.

CAUTION

Computer should not be connected to a power line serving heavy equipment which may generate excessive voltage fluctuations.

The line regulator, on LPS-2 below Diode Board No. 2, is a Stabiline Type IE5202R Voltage Regulator which provides +1 percent regulation of the 230-volt a-c input to the d-c power supplies (see Figure I-6). It will operate over

- | | |
|-----------------------------|--|
| 1. Top Panel | 15. LPS-3 (+225 v Unregulated) |
| 2. Fan Shutters | 16. Terminal Strips |
| 3. Fan Assembly | 17. LPS-4 (Relay Sequencing) |
| 4. Fan Connector Panel | 18. Front Panel |
| 5. Upper Frame Assembly | 19. Contactor Panel |
| 6. UTC-1 (IBM Thyratrons) | 20. Fuse Panel |
| 7. UPS-3 (+225 v Regulator) | 21. LPS-2 (Stabiline Supply) |
| 8. Rear Panel | 22. Drum Motor Assembly |
| 9. Right Side Panel | 23. Drum Barrel Assembly |
| 10. Diode Board | 24. Memory Tube Chassis |
| 11. Tube Chassis | 25. Left Side Panel |
| 12. Lower Frame Assembly | 26. Computer Test Panel |
| 13. LPS-1 (Filament Supply) | 27. UPS-4 (-300 v Supply) |
| 14. Cable Connector Panel | 28. UTC-2 (Fill and Flexo-
writer Relays) |

LEGEND FOR FIGURE I-2

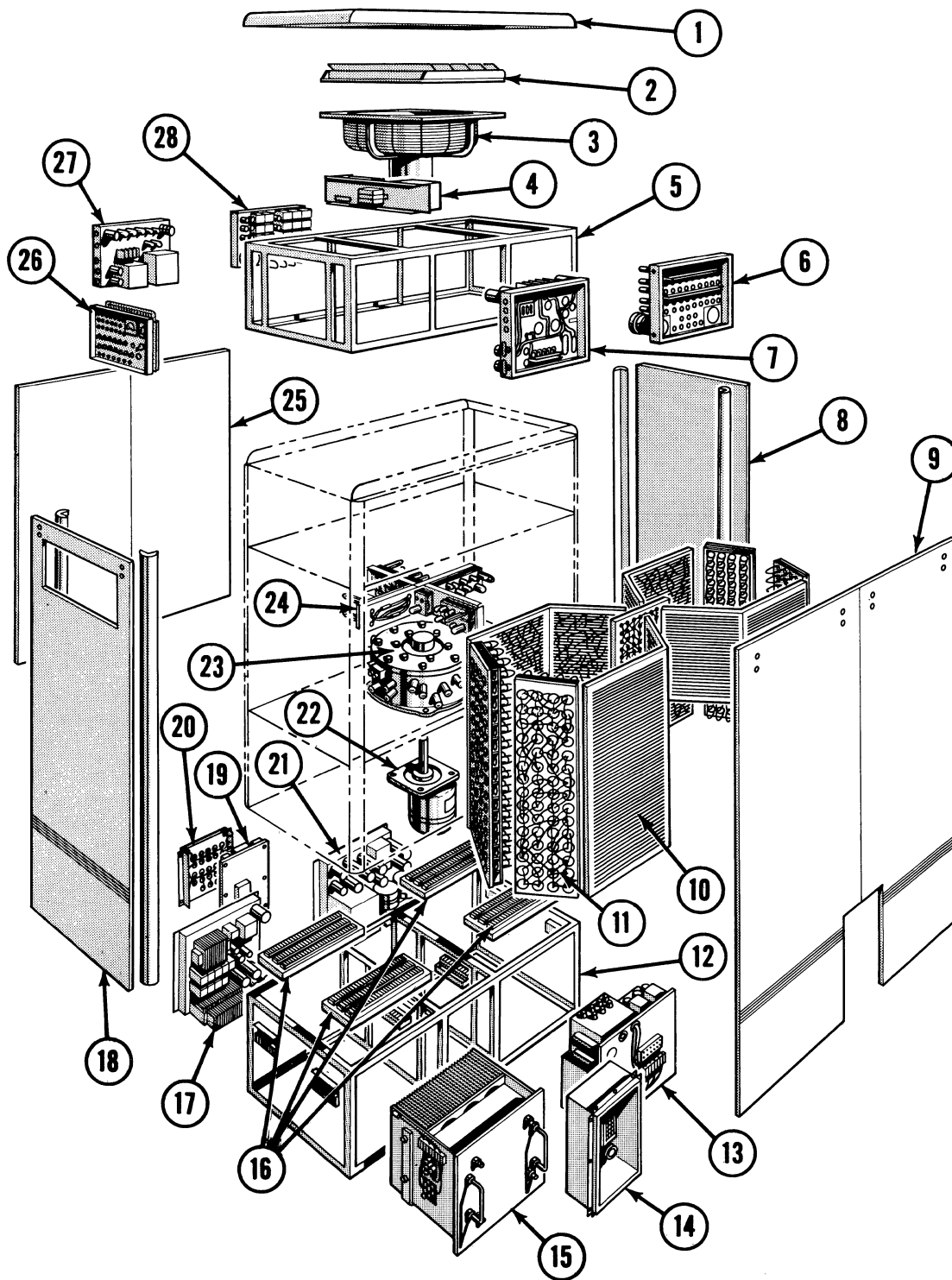


Figure I-2. Model 102-A General Purpose Computer (exploded view)

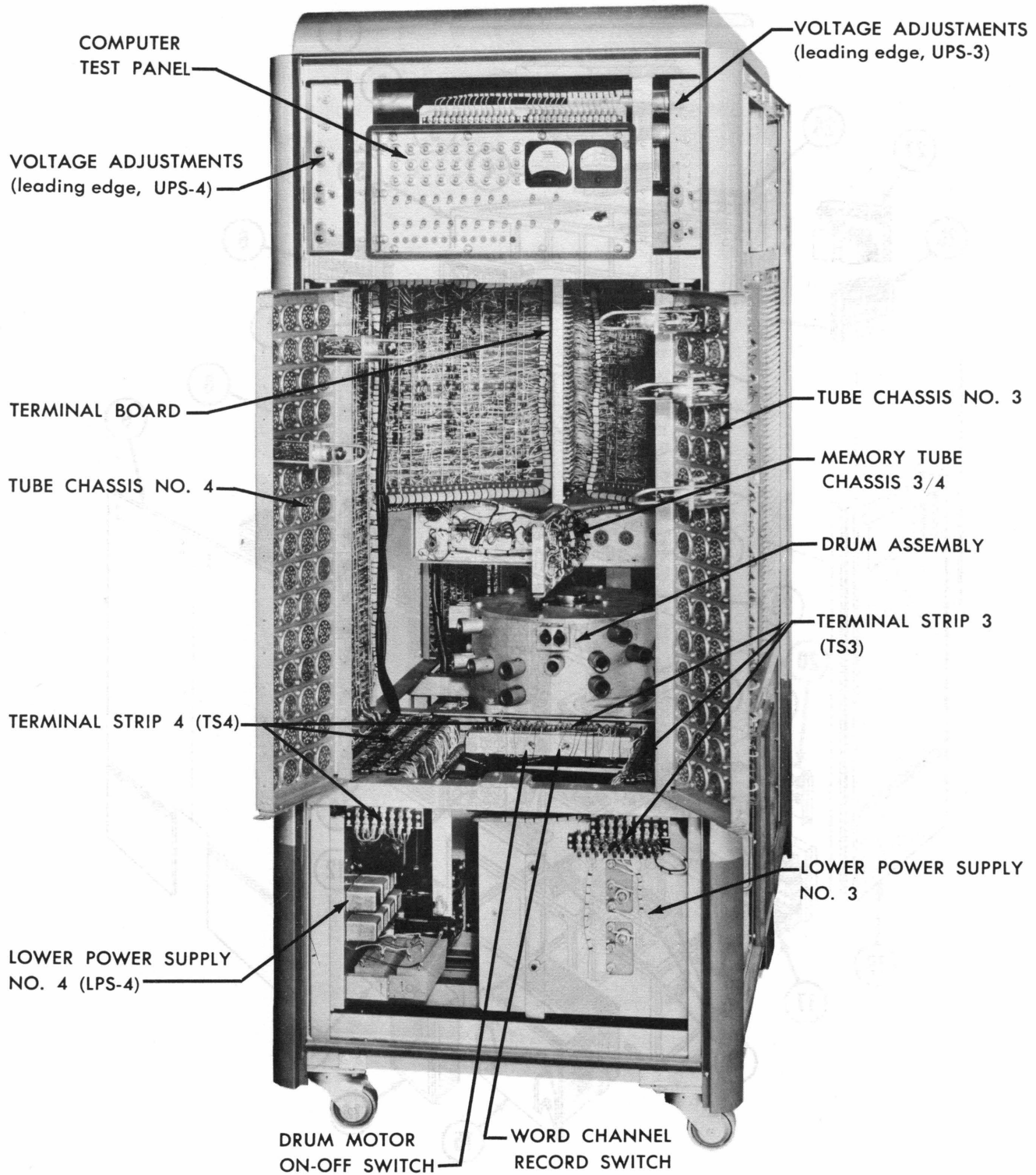


Figure I-3. Model 102-A General Purpose Computer (front view)

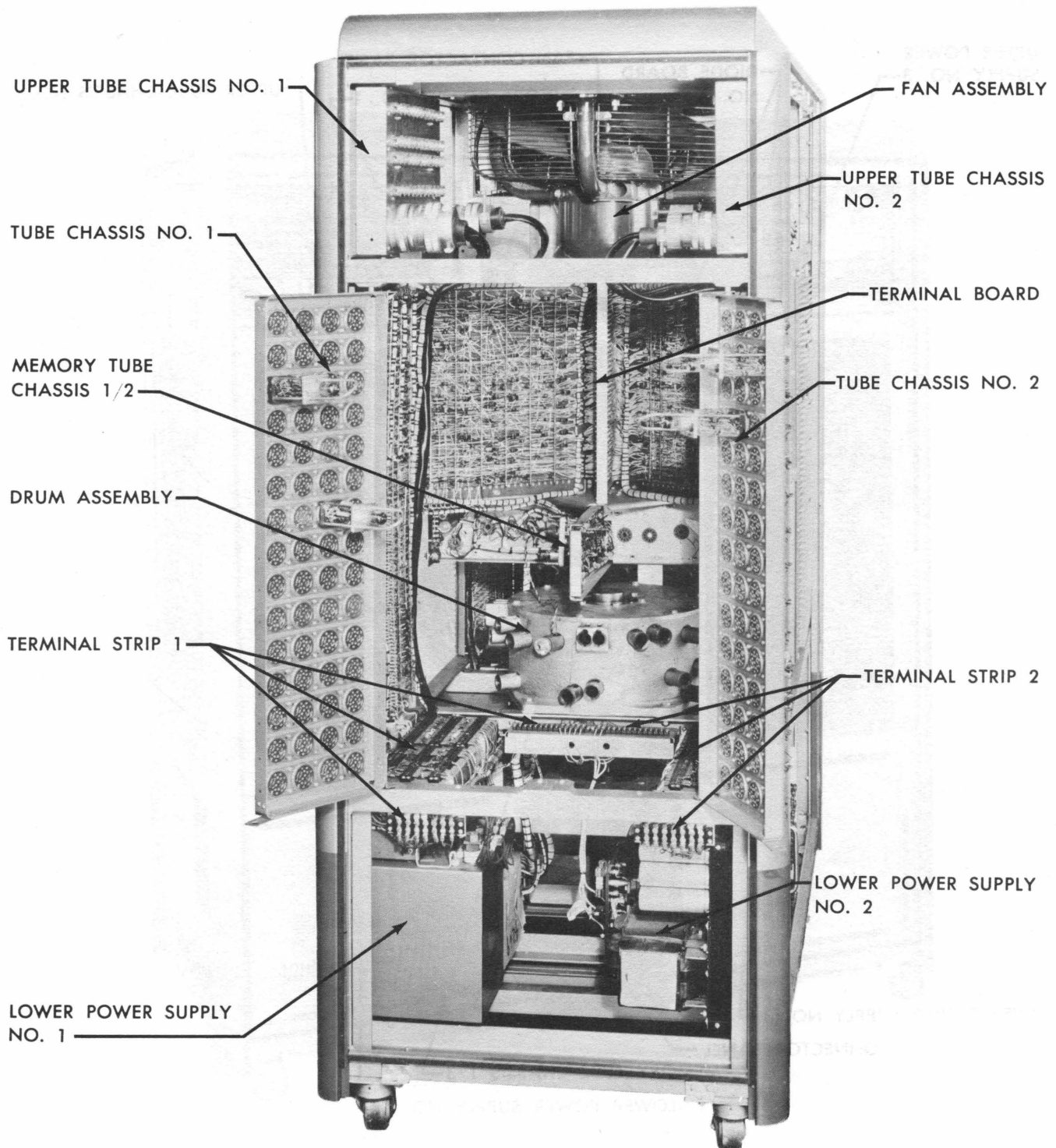


Figure I-4. Model 102-A General Purpose Computer (rear view)

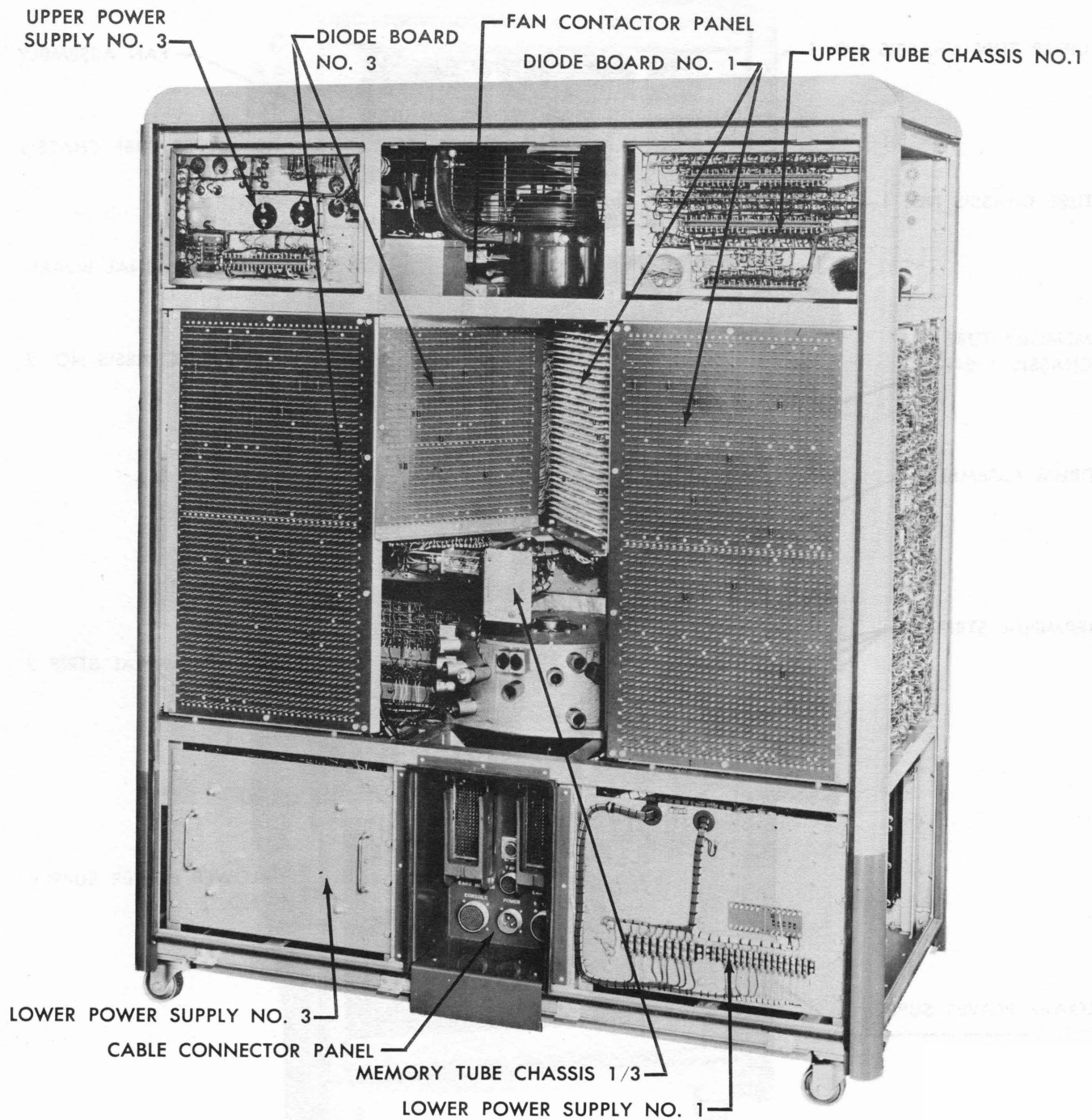


Figure I-5. Model 102-A General Purpose Computer (right side)

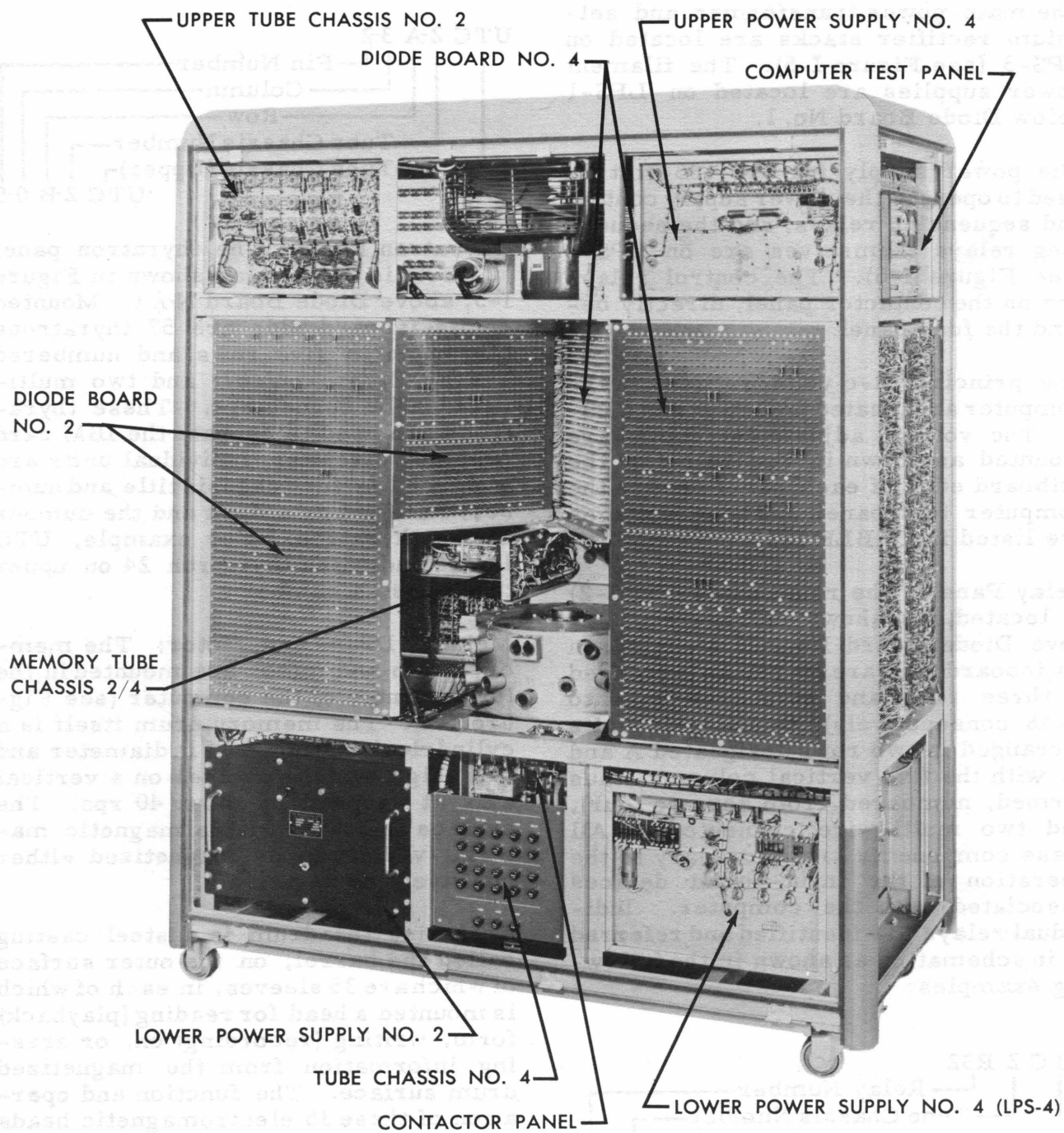


Figure I-6. Model 102-A General Purpose Computer (left side)

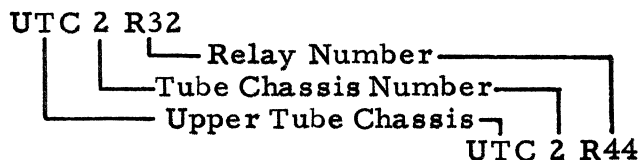
a range of 195 to 255 volts ac.

The main power transformer and selenium rectifier stacks are located on LPS-3 (see Figure I-5). The filament power supplies are located on LPS-1 below Diode Board No. 1.

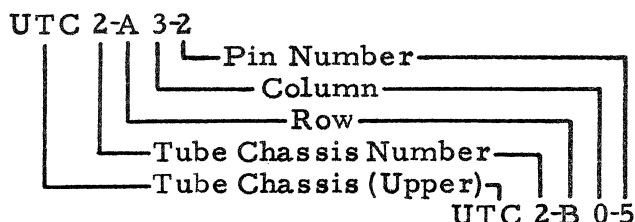
The power supply for the +85-volt dc used to operate the power supply control and sequencing relays, and the sequencing relays themselves are on LPS-4 (see Figure I-6). The control relays are on the contactor panel, directly behind the fuse panel.

The principal d-c voltages used in the computer are located on UPS-3 and UPS-4. The voltage adjustment points are mounted as shown in Figure I-3 on the outboard edge of each panel next to the computer test panel. These voltages are listed in TABLE I-3.

Relay Panel: The relay panel (UTC-2) is located, as shown in Figure I-6, above Diode Board No. 2. Mounted on its inboard side are 24 relays (arranged in three rows and numbered R-25 to R-48 consecutively), ten plug-in units (arranged in two rows designated A and B, with the five vertical columns, thus formed, numbered from zero to four), and two multi-wire connectors. All these components are necessary to the operation of the input-output devices associated with the computer. Individual relays are identified and referred to in schematics as shown in the following examples:



The individual plug-ins are located as shown below:



Thyratron Panel: The thyratron panel (UTC-1) is located, as shown in Figure I-5, above Diode Board No. 1. Mounted on its inboard side are 57 thyratrons (arranged in five rows and numbered V-1, V-2, ... V-57) and two multi-wire cable connectors. These thyratrons are used to operate the IBM card punching circuits. Individual units are identified by tube chassis title and number, followed by a dash and the number of the thyratron. For example, UTC 1-V24 indicates thyratron 24 on upper tube chassis number 1.

Memory Drum and Motor: The memory drum and motor are mounted in the lower center of the computer (see Figure I-2). The memory drum itself is a cylindrical casting, 12" in diameter and 5.3" high, which revolves on a vertical axis at a speed of about 40 rps. The surface is coated with a magnetic material which can be magnetized either positively or negatively.

Enclosing the drum is a steel casting called the barrel, on the outer surface of which are 35 sleeves, in each of which is mounted a head for reading (playback) form, writing (recording) on, or erasing information from the magnetized drum surface. The function and operation of these 35 electromagnetic heads are described in detail in SECTION IV.

The electric motor used to drive the memory drum is mounted under the drum. It is a 1/2 hp, 60-cycle, single phase, 115/230 volt Master Gearmotor,

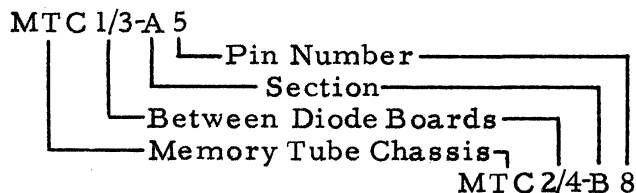
D - C VOLTAGE	APPLICATION IN COMPUTER
+ 415 V	Plates of Memory record tubes.
+ 250 unregulated	Plates of main memory play-back amplifiers.
+ 225 regulated	Plates of most tubes, logical product resistors, and source of numerous other voltages including 150, 125, 100 regulated, 115 volts. It is obtained from the 250V unregulated.
+ 180 } + 170 } + 150 } + 115 } + 125 }	obtained from +250V unregulated
	Clamping diodes of memory limiter amplifier input.
	Clamping diodes of record tube driver.
+ 100	Upper clamping diodes; true logical voltage level for logical sums and products; recirculating line record tube cathodes.
+ 20 } + 10 }	Lower clamping diodes (clamped flip-flops, flip-flop drivers, logical drivers, clamped one-shot, clipper; false logical voltage level.
	Upper and lower clamp bias for the memory clipper.
+ 6 from +100V	Noise clipper bias.
- 8 from -300	Grid clamp bias.
- 10 from -300	Thyratron grid bias.
- 300	Grids or cathodes of numerous tubes; source of negative bias voltages.

TABLE I-3. Principal D-C Voltages

drawing 3.5 amp at 230 volts, and has a shaft speed of 2430 rpm.

Test Panel: The computer test panel is placed at the forward end of the computer cabinet, above Tube Chassis 3 and 4 (see Figure I-3). This panel, along with the smaller voltage adjustment panels to the right and left, contains all the meters, switches, indicator lights, test jacks, and adjusting potentiometers necessary to enable operators and maintenance engineers to check the operation of the computer at all times, and to manually control various operating routines of the computer for test purposes. The various functions of these controls and indicators are explained in detail in SECTION V.

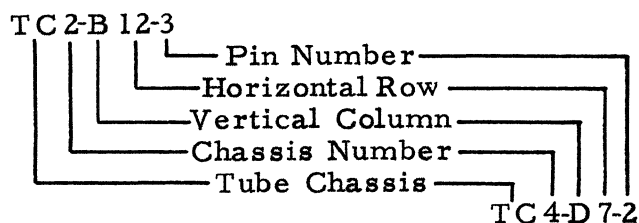
Memory Tube Chassis: The four Memory Tube Chassis are mounted above the memory drum (see Figure I-2). These chassis contain plug-in units and associated wiring for circuitry most closely associated with the electromagnetic heads mounted in the memory drum barrel. The individual chassis are identified with respect to the two closest diode boards. Thus, MTC 1/2 would indicate the memory tube chassis mounted between Diode Boards 1 and 2. Each chassis is broken into sections, labeled A, B, C, etc. Units mounted on the chassis are referred to as shown in the examples below:



Tube Chassis: There are four hinged tube chassis, two at each end of the computer, on which are mounted the majority of the plug-in units and associated wiring (see Figures I-3 and I-4). The chassis are numbered one through four, the number corresponding to the

nearest diode board. Each chassis contains 60 plug-in unit sockets arranged in four vertical columns and so mounted that the plug-in units face inboard with the chassis closed while the terminal points of the sockets are conveniently exposed, for test purposes, on the outboard side of the chassis. All electrical connections from associated circuitry are made to these numbered socket terminals. Since each chassis is hinged to swing outboard, ready access is also provided to the plug-in units.

The four vertical columns on each chassis are designated A, B, C, D; and the horizontal rows are numbered from 1 to 15 down the side. The number and letter grid permits quick location of the desired plug-in unit as illustrated in the example below:

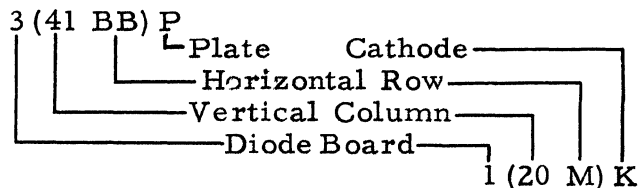


Diode Boards: The locations of the four diode boards housed in the computer are shown in Figure I-2. Actually each diode board consists of two parts, the large board mounted parallel to the side of the computer and a smaller board which angles inward toward the center of the machine. These boards provide space for the logical networks of diodes and resistors.

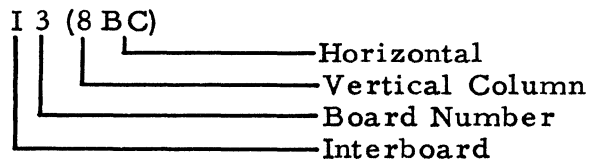
Each diode board is equipped with 2600 diode clips arranged in pairs (see Figures I-5 and I-6). This shank of each diode clip projects through the board to the reverse side to connect to the resistors and wiring of the logical networks in which the diodes are utilized. A diode is retained in a pair of clips, thus each board has a capacity of 1300 diodes. The upper clip of each pair of

diode clips is for connection to the anode (plate), and the lower to the cathode.

The 30 vertical columns on the larger section of each diode board are numbered from 25 through 54, and the horizontal rows are lettered A through DD. The 25 vertical columns on the smaller boards are numbered from 0 to 24, and the 16 horizontal rows are lettered A through O. Diodes, and consequently, each pair of diode clips are identified in the following manner:

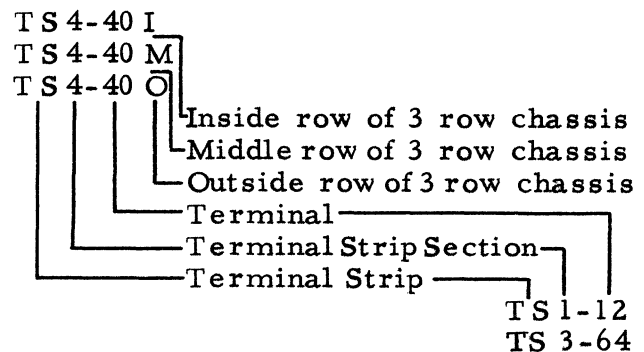


Interboard: An interboard placed above the memory tube chassis and below the fan, runs lengthwise inside the computer between the smaller wings of the four diode boards, and provides 760 terminals for connection between the boards. This interboard is arranged in four quadrants, two on each side of the board; and each quadrant is gridded in 10 columns from 1 through 10 and 38 rows lettered AA through OB. Each quadrant is numbered, one through four, according to the diode board which connects to it. Reference to points on the interboard are made as shown below:



Terminal Strips: There are four main terminal strip sections, numbered 1 through 4 corresponding to the nearest diode board. Each section consists of three separate chassis mounted as shown in Figures I-3 and I-4. One-half of the single row chassis, mounted beneath Memory Tube Chassis 3/4 and

1/2, is associated with each section; and the terminals on each half are numbered 1 through 13. The four three-row chassis are mounted behind and beneath the four diode boards. The terminals on each row are numbered 20 through 60 and designated by the letter "O" (outside row), the letter "M" (middle row), and the letter "I" (inside row). Each of the four single row chassis is mounted on the frame beneath the associated tube chassis and their terminals are numbered 61 through 67. The following examples illustrate how individual terminals are referenced in the schematics and located on the terminal strips:



Fuse Panel: The fuse panel, containing most of the principal fuses used in the computer, is located as shown in Figure I-6. The twenty-three fuses are grouped by function and clearly labeled. The two main power line fuses are mounted on the contactor panel, located directly behind the fuse panel.

Cable Connector Panel: The cable connector panel is located as shown in Figure I-5. All interconnections between the computer and its input power, operator's console, tape unit or units, and IBM punch card machines are made to this panel. A spare tape plug is also provided.

Fan and Shutters: An 18" fan, with guard and shutters of the same size, is located as shown in Figure I-2. There are separate motors to operate the fan

and shutters. The two thermostats which automatically control the operation of the fan and shutter motors are located on the fan connector panel, also shown in Figure I-2.

Operator's Console

The operator's console is a metal desk of conventional design on which has been mounted the computer control panel and

the Flexowriter. Circuitry for the computer control panel is contained under the hinged desk top to the right of the Flexowriter. The console is connected to and operates off the computer's power supply. Although the Flexowriter also connects to the computer through the operator's console, it contains its own 90-volt d-c power supply that is energized from a standard 115-v a-c outlet.

Installation

GENERAL INFORMATION

All 102-A computing system installations will be made by NCR Service personnel. For this reason, the following text does not describe in detail the physical installation; but, rather, establishes requirements and system layout that must be provided to insure proper operation, ease of maintenance, and adequate filing and storage space. The weight, dimensions, and power requirements of the Model 102-A computing system have been listed in TABLES I-1 and I-2.

PRELIMINARY REQUIREMENTS

Filing and Storage

Filing and storage considerations will involve space for maintenance equipment and spare parts, media stock and active media, and a library of reference data.

1) Maintenance Storage Requirements:

Spare Parts -- 8 cubic feet.

Test Equipment and Tools -- 4 cubic feet.

Prints and Manuals -- 8 1/2" x 20".

2) Media Stock Storage: Storage space requirements for Flexowriter paper tapes, magnetic tapes, and IBM cards will vary widely, depending on the specific customer applications. The following dimensions are given as an aid to estimating space requirements:

ITEM	QUANTITY	DIMENSIONS
Paper Tape	1 roll (1000')	8" dia. x 1" thick
Mag. Tape	1 reel (1200')	10 1/2" dia. x 1 1/4" thick
IBM Cards	40,000	18" x 27" x 48"

3) Filing Active Media: Short lengths of punched Flexowriter paper tape are best stored coiled in cardboard boxes 4" x 4" x 1", available from Commercial Controls Corporation. Reels of 126 magnetic tape are best stored in 35 mm motion picture film cans, available from large photographic equipment dealers. For long term storage, a year or more, a dry cool storage environment is recommended.

Standard file cabinets with space for 40,000 or more IBM cards are available from Peerless Steel Equipment Co., Philadelphia, Pa., and Corry-Jamestown Mfg. Co., Corry, Pa.

4) Conventional Library and Filing: This requirement will vary widely with the specific customer applications but should include space for programming manuals, program flow diagrams, etc.

Thermal

The recommended maximum room ambient temperature for operation of a 102-A computing system is 90° Fahrenheit. For example, a system using a Model 102-A, two tape units, and two IBM punches would liberate 35,000 BTU/hr. Adequate ventilation and/or air conditioning should be provided in circumstances where it appears that heat generation on this order in the computing area would cause the room ambient to rise above 90° F. TABLE II-1 lists the heat generated by the various units in this computing system.

It is recommended that the computer not be located in a dead air space (alcove, separate room, etc.) unless an overhead vent for exhaust air and floor level intake vents are provided. Free air flow into the space between the bottom of the computer cabinet and the floor on all four sides of the cabinet should be unimpeded.

When operating the computer with side panels off for maintenance purposes, a 16" fan should be placed at each end of the computer to maintain a safe operating temperature inside the computer.

SYSTEM LAYOUT

Although size and shape of the available space and amount of equipment to be used will dictate to a considerable extent system layout, the following recommendations for locating and spacing the units will tend to increase the efficiency and ease of operation and maintenance of a 102-A system. The interconnection and power cabling of the units in the system are listed in TABLE II-2.

All power should be brought into the installation through circuit breakers, with the load divided as shown in the last column of TABLE I-2. The circuit

breaker cabinet may be placed wherever is most convenient.

UNIT	MAX BTU/HR
Model 102-A	22,000
Console and Flexowriter	1,500
Model 126	3,500
IBM 523	2,400

TABLE II-1. Thermal Characteristics

CAUTION

Computer should not be connected to a power line serving heavy equipment which might generate excessive voltage fluctuations.

A typical system layout is illustrated in Figure II-1. This system includes the Model 102-A Computer, four Model 126 Magnetic Tape Handling Units, and two IBM 523 punches. Storage space has been provided for 48 reels of magnetic tape (well over five million words capacity), 80,000 IBM cards (approximately 320,000 words), sufficient space for maintenance equipment and spare parts, and a cabinet for miscellaneous filing. Total space required for this layout is 13'6" x 16', which is about minimum for a system of this size.

Note that in the suggested layout, the operator is so placed as to be able to observe the lights on the Test Panel on the computer. This arrangement is desirable for convenience in running test problems.

About three feet of space are provided on all sides of the computer to enable the maintenance engineer to use test equipment wherever required.

The tape units are placed with the long side accessible for changing tapes, and so that the operator can observe the

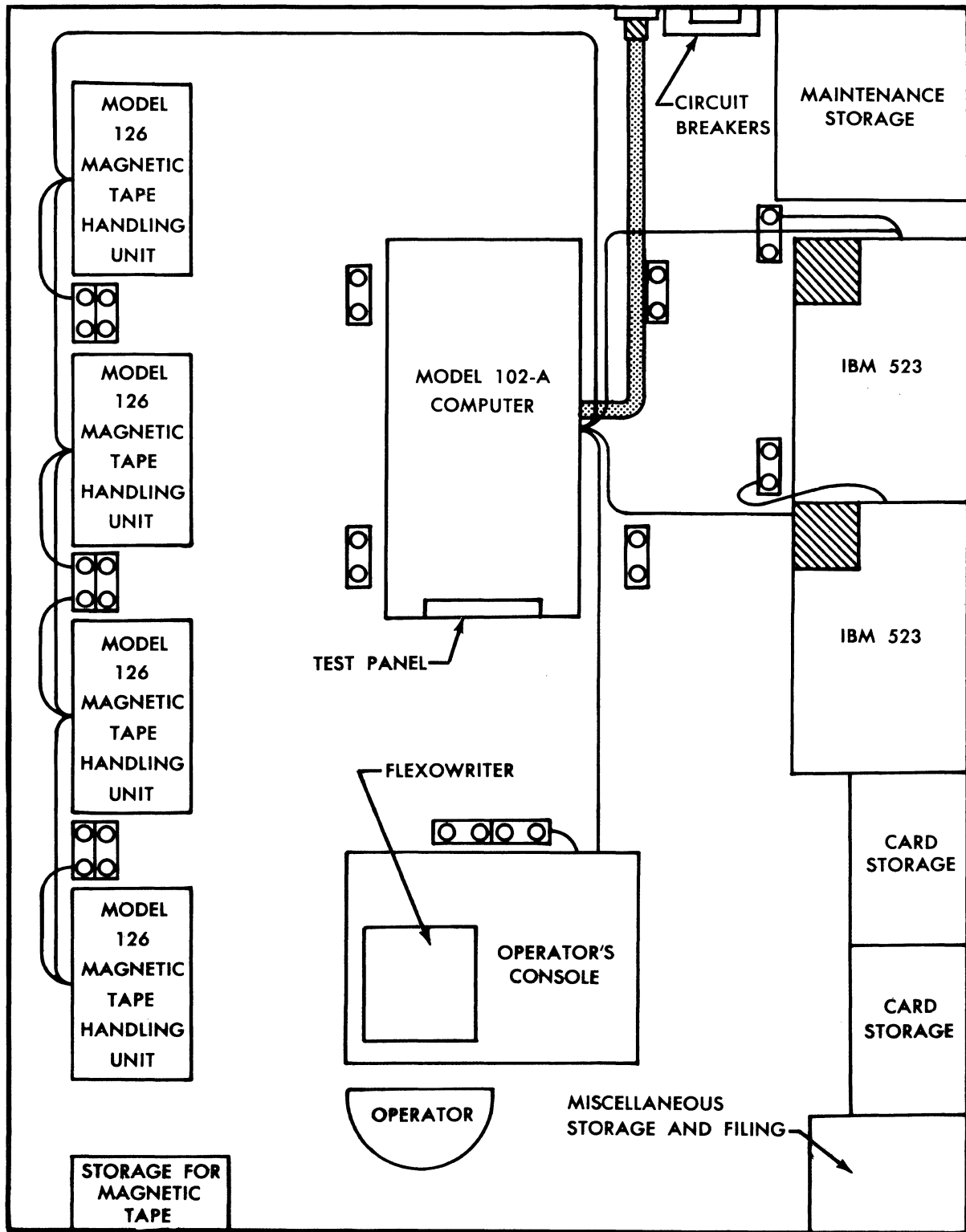


Figure II-1. Typical System Layout

	Length	Cable Dia.	Power Plug	Mating Receptacle
Computer Power	20'	1"	Hubbell 7388	Hubbell 7380
Console Power (see Flexowriter)				
Flexowriter Power	8'	3/8"	Standard 115V	(with ground lead)
Tape Unit Power	10'	3/8"	Hubbell 7567	Hubbell 7580
IBM 523 Power	7'	1/2"	Standard 115V	(with ground lead)
Computer-Console	20'	1 1/2"	Provided with System	
IBM-Computer	8'	1 1/2"	"	"
Tape Unit to Tape Unit	5'	1 1/2"	"	"
Tape Unit to Computer (2)	15'	1 1/2"	"	"

TABLE II-2. Interconnection and Power Cabling

action of the tape drive mechanisms.

SYSTEM'S RECORDS

Beginning with the initial installation, a complete record of operation and maintenance should be kept on your 102-A system. To accomplish this, the following items are furnished with the system: Log Book, System Operation Report Sheet(s), and Component Check Sheet(s). In addition, service engineers of The National Cash Register Company prepare for your records detailed Service Reports, covering each preventive maintenance call.

The log book will enable the customer to keep a daily record of system operation. This daily record will then be summarized on the weekly System Operation Report Sheet which is prepared by

a service engineer and signed by the customer. This report, filled out in duplicate, is a breakdown of up-time and down-time on the basis of the scheduled work week. The Component Check Sheet lists the components in convenient groups for the purpose of periodic routine maintenance checking. This record enables the customer to more or less anticipate component failures and thus prevent them. This check sheet along with the System Operation Report Sheet will be furnished by our Service Department.

Such records, if kept up to date, will permit more accurate estimates of computer availability for work scheduling, improved preventive maintenance schedules, and more rapid trouble diagnosis and repair.

Functional Specifications

INTRODUCTION

The Functional Specifications for the Model 102-A computer are included in this manual to acquaint the reader with the configuration of data within the computer, the operation and components of the memory, and the commands used for all computing system operations. An understanding of these unique features of the Model 102-A computer will enable the reader to follow the more detailed explanations of computer theory that appear in SECTION IV. It is further recommended that if the reader is unfamiliar with the binary and octal number systems, referred to throughout the text, he should read at this time the discussion under "Number Systems and Arithmetic Processes" in the APPENDIX.

BASIC PRESENTATION OF DATA

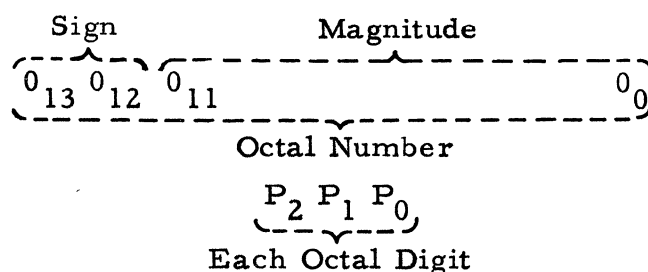
Word Structure

A word in the 102-A computer consists of 14 octal digits (0_0 through 0_{13}), each of which is composed of three binary digits (P_0 , P_1 , and P_2). Thus, each word contains 42 binary digits (bits), arranged in the time sequence from $P_0 O_0$ through $P_2 O_{13}$.

Configuration of Octal Numbers

A word may be either an octal number or a command. An octal number is composed of two sign digits and 12 magnitude digits,* with the octal point placed between the sign and magnitude digits.

This configuration is as follows:



Digits 0_0 through 0_{11} represent the magnitude of the number, and the discussion in this manual assumes that the octal point is located between 0_{11} and 0_{12} . Thus, all numbers are considered as octal fractions lying between -1 and +1. However, it is possible in certain arithmetic computer operations such as addition, subtraction, and double-length multiplication to place the octal point elsewhere in the word.

The sign convention for octal numbers is shown in Table III-1.

Octal Sign Digits	Binary Equivalent	Interpretation
00	000 000	Positive, no overflow
01	000 001	Positive, overflow
02	000 010	Negative, no overflow
03	000 011	Negative, overflow

TABLE III-1 Octal Number Sign Code

Looking at the six bits of the two sign digits, the four most significant bits are always zero. The next bit, $P_1 O_{12}$, indicates the algebraic sign of the octal number: zero, if positive; one, when negative.

The least significant bit, $P_0 O_{12}$, is an

*Hereafter in this manual, the word "digit" by itself will mean octal digit.

overflow marker: zero, no overflow; one, overflow present. Any arithmetic operation in the computer which would have yielded a result ≥ 1 will cause this overflow marker to appear.

Configuration of Decimal Numbers

A decimal number results in a word with a different configuration as follows:

Sign Magnitude
(6 bits) (9 decimal digits, each of 4 bits)

The sign of a decimal number is expressed in the six most significant bits of the word. The two most significant of those six bits are always zero and are omitted from TABLE III-2; the $P_1 O_{12}$ bit always contains the algebraic sign, and the $P_0 O_{12}$ bit provides overflow information as shown:

Decimal Sign Digit		Binary	Interpretation
Filling	Printing	Equivalent	
+ or 0	+	0000	Positive, no overflow
- or 2	-	0010	Negative, no overflow
1	p	0001	Positive, overflow
3	n	0011	Negative, overflow
Space bar	Space	1100	Suppress sign printing

TABLE III-2. Decimal Number Sign Code

The decimal digits of the magnitude are entered into the computer, or printed out, by means of a four-bit binary code which includes two special characters used in editing output data. This is shown in Table III-3.

As previously mentioned, the computer cannot use these binary-coded decimal numbers directly. For example, suppose the last two decimal digits in a number are 79. In binary-coded decimal, this looks like 0111 1001. As a binary number, this latter has the decimal value of 121, not 79. A suitable decimal-to-octal conversion routine must be included in all programs using

decimal numbers at the input, or where a decimal output is desired. Thus, the decimal word is converted to the octal number word previously shown.

Decimal Digit	Special Character	Binary-Coded Equivalent
0		0000
1		0001
2		0010
3		0011
4		0100
5		0101
6		0110
7		0111
8		1000
9		1001
	Space	1100
	Period	1111

TABLE III-3. Binary Code for Filling and Printing Decimal Digits.

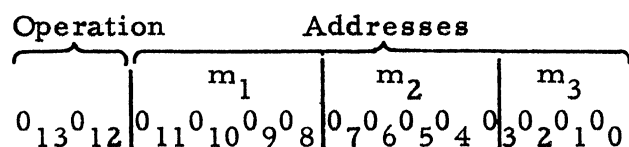
Computer Program

A program consists of a list of commands which are entered into the computer to accomplish the solution of a certain problem, or some series of operations on a set of data. The commands are arranged in proper sequence by the programmer.

Configuration of Commands

While commands are always in binary form within the computer, they are abbreviated to octal numbers outside the machine. Each command consists of the following: (1) a two-digit number called the Operation, designating the arithmetic, logical, or input-output function to be performed; and (2) three four-digit numbers (m_1 , m_2 and m_3) which provide the addresses of cells in the memory where information is stored. (In some cases, as will be seen later in the discussion of the PRINT and MAGNETIC TAPE UNIT commands, m_1 , m_2 , or m_3 may contain special in-

formation rather than an address.)



Example: A typical command would be typed on the Flexowriter as
35 0123 0134 0255

This command tells the computer to look up the number in cell 0123 (contents of address m_1) and look up the number in cell 0134 (m_2); perform Operation 35, which means adding the two numbers; then put the result in cell 0255 which is m_3 . Note: One important conventional notation is that parenthesis is used around an address to denote the contents of an address. It is important to distinguish between the address m_1 , and the word stored at that address, (m_1) .

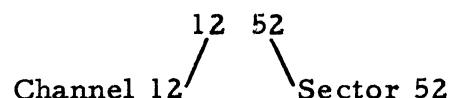
Because most commands refer to three cells or addresses during their execution, the 102-A is called a three address computer.

MEMORY

MAIN MEMORY

The main memory, designated by the capital letter M, contains 16 parallel circular channels, each of which goes around the circumference of the memory drum. Each channel contains 64 words, thereby providing a main memory capacity of $64 \times 16 = 1024$ words. The main memory channels are numbered octally from 00 to 17, corresponding to the first two digits of an address. Each channel is divided into 64 sectors. See Figure III-1. A sector is one word in length. Sectors are numbered octally from 00 to 77. The location on the drum corresponding to an address is called a cell, and its position is specified in the main memory by channel and sector as follows:

Typical main memory address:



Information is read from, or written (recorded) on, the cells of the main memory by means of combination read-write heads. There is a read-write head for each of the 16 channels. How these heads and other electronic circuitry are used in finding addresses is described in detail in SECTION IV.

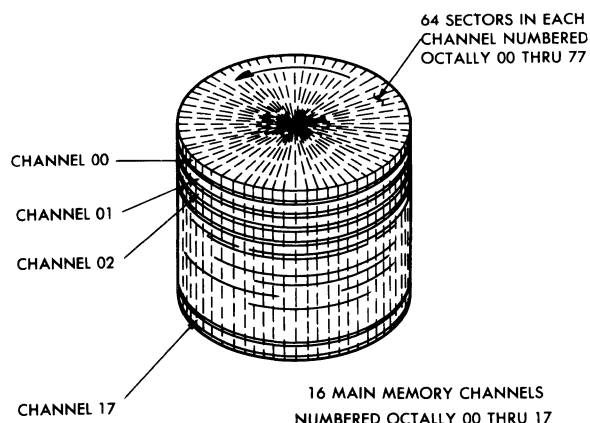


Figure III-1. Memory Channel and Sector Designations

RECIRCULATING REGISTERS

The Buffer Register, identified as J, is a recirculating line with a content of eight words. As shown in Figure III-2, the buffer has separate heads for reading and writing, spaced 334 pulses apart on the drum circumference. Information read off the drum by the read head is re-recorded on the drum by the write head, and this material therefore is recirculated until the line is broken for the insertion of new information. However, the eight-word content of the buffer can be read out for computation purposes without affecting the recirculation. Addresses in the J (buffer) register are numbered octally from 2000 through 2007.

There are also four one-word recircu-

lating registers, identified as E, F, G, and H. Each of these registers has a read head and a write head, spaced approximately one word apart, which operate similar to those in the Buffer Register. The E and F registers are used for arithmetic purposes, to store operands and results of operations. H, called the control register, is used to contain the operation number, the addresses of operands, and the address of the next command. The G register is used occasionally for control purposes and at other times for various arithmetic and logical purposes.

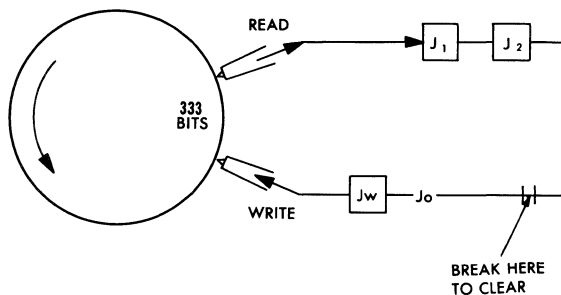


Figure III-2. Buffer Register

During computation, operation of the E, F, G, and H registers is automatic, and the programmer cannot use them for storage.

It should be noted that storage in all the recirculating registers, E, F, G, H, and J, is volatile. If the computer power is turned off, their content is immediately lost. Storage in the main memory channels is permanent in the sense that, once recorded, information will remain on the drum until it is intentionally replaced. Recording new information in a main memory cell simultaneously removes the previous contents of that cell, as is typical of magnetic recording devices.

WORD CHANNEL

The word channel on the memory drum operates to identify the sectors of the

main memory channels and the buffer. Called M_w , the word channel contains 64 words of the octal form

00 0000 0000 0000 to 00 0077 0077 0077

The last two digits of the address specify the sector. These last two digits are compared to the output of the word channel read circuits. When the word channel digits coincide with the sector digits in the address, the next word appearing in the selected channel is read out, or any desired word is written into this selected memory cell.

SPECIAL ADDRESSES

Special addresses, that is, addresses not in the main memory, include cells 2000 through 2007 in the buffer register, J. Whenever the first two digits of an address are 20, the buffer is selected instead of a main memory channel. Then comparison is made between only the last digit of the word channel and the last digit of the desired address. At coincidence, connection is made between the E or F register and the buffer output or input circuits.

Cell 2100 is another special address, which may be used in the m_1 or m_2 (look-up) positions in H, but not in the m_3 (putaway) position. When 2100 appears as the digits on m_1 or m_2 , the E or F register will be filled automatically with zeros. Cell 2100 does not exist physically, since filling with zeros is actually accomplished electronically by breaking the E or F line for one word time. Therefore, there can be no putaway to 2100. If 2100 is put into m_3 of H, the 1 as the second digit of the address will be ignored, and the putaway will be in cell 2000 of the buffer.

Cell 3000 also has a special meaning. If this cell is called for in m_1 of H, the contents of the G register are transferred to the E line. If 3000 is in m_2 of H, the contents of G go into F. With

3000 in m_3 of H, putaway is in buffer cell 2000 as for cell 2100. There is no putaway in cell 3000.

To summarize, when writing into the memory (putaway), only the main memory channels and buffer register may be used as storage space, a total of 1032 decimal addresses. In look-up, or reading from the memory, not only the main memory and buffer may be used, but also special cells 2100 and 3000.

CHANNEL SELECTION

Channel selection is accomplished electronically by means of appropriate logical circuitry associated with the flip-flops L_1 through L_5 which are set up to correspond to the least significant five bits contained in the most significant two octal digits of an address.

Access to the Memory

Access to the memory is achieved according to the following procedure. Consider the sample command 35 0123 0134 0255, which means "Add the contents of cell 0123 to the contents of 0134 and put sum in 0255".

In the execution of all such arithmetic commands, the command is read into the H register. Then the contents of m_1 of H is read into the E register, (m_2) of H into F, the Operation (add, divide, etc.) is carried out, with the result appearing in E and replacing the operand, which is no longer needed. The new contents of E (result) are read into the cell called out in m_3 of H.

In this example, m_1 of H is 0123. Channel 01 is selected electronically, and sector 23 is located by means of the word channel, M. Now cell 0123 is connected to the E line for one word time, so that its contents are read into E. Similarly, cell 0134 is found and its contents are read into F. Then the

contents of the E and F registers are added, with the results going into E and replacing its previous content (m_1). Finally, the E line is connected to cell 0255, which has been found in the usual way, so that the answer can be recorded in that cell as the command specified.

Access Time

Access time depends upon several factors. The memory drum rotates at approximately 40 rps, and thus requires 25 milliseconds per revolution. If a desired word has just passed a main memory channel read-write head when sector look-up begins, a full revolution will be required before the information can be transferred. Thus, maximum access time for the main memory is 25 ms. Mean random access time is therefore half this amount, or 12.5 ms. For the buffer, maximum access time is $25/8 = 3.125$ ms.; mean random access time = 1.56 ms.

Where computing time must be minimized, it is necessary occasionally to re-number the word channel for minimum-access coding. Re-numbering of M^w can be accomplished by using the Flexowriter, or paper tape, and a special circuit built into the computer.

TIMING CHANNELS

Timing channels on the memory drum include a working clock channel, C_1 , and a duplicate, C_2 , both of which are permanently recorded and are identical. The duplicate is provided as a safeguard in case C_1 is damaged in any way and is used only in that contingency. On each of these clock channels, there are $64 \times 42 = 2688$ clock pulses, each corresponding to one bit-space in all channels of the drum. Output of the clock playback (read) circuit is a square wave of 64×42 pulses $\times$ 40 rps = 107,520 cps frequency (approximately). The factor of 40 rps is subject to slight variations caused by changes in power-line fre-

Operation	Instr & Code	m ₁	m ₂	m ₃	Remarks															
- - - ADDRESSES OF - - -																				
Add	ad - 35	Augend	Addend	Sum	{ If (m ₁) is a command, the I portion is retained and (m ₂) is treated as a number															
Subtract	su - 36	Minuend	Subtrahend	Difference																
Multiply-Round	mr - 25	Multiplier	Multiplicand	Product	Product is rounded															
Multiply-Double	md - 26	Multiplier	Multiplicand	LSH of Product	MSH of Product → Cell following m ₃															
Divide-Round	dr - 23	Dividend	Divisor	Quotient	Quotient is rounded															
Divide-Double	dd - 24	Dividend	Divisor	Remainder	Unrounded quotient → Cell following m ₃															
Shift Logical	sl - 27	Operand	Shift control	Result	{ Magnitude of (m ₂) = No. of binary shifts I of (m ₂): + = Shift left - = Shift right															
Shift Magnitude	sm - 30	Operand	Shift control	Result																
Scale Factor	sf - 31	Control	Operand	(m ₁) - # shifts	Shifted operand → Cell following m ₃															
Extract	ex - 32	Source-word	Extractor	Cell Modified	Bits of (m ₁) → (m ₃) per "1" bits of (m ₂)															
Test Overflow	to - 37	Tested Word	2100 or 3000	Alternate Cmnd.	If overflow bit, execute alternate command															
Test Magnitude	tm - 34	Magnitude ₁	Magnitude ₂	Alternate Cmnd.	If Mag ₁ > Mag ₂ , execute alternate command															
Test Algebraic	ta - 33	Word ₁	Word ₂	Alternate Cmnd.	If Word ₁ > Word ₂ , execute alternate command															
Test Switch	ts - 17	Switch No.	2100 or 3000	Alternate Cmnd.	If switch is up, execute alternate command															
Test Search	ts - 17	2400	2100 or 3000	Alternate Cmnd.	If "bs" still in progress, execute alternate command															
Print	pr - 21	First Word to be printed	Print control	No. of Words (not an address)	<table border="1"><tr><td>No</td><td>I of (m₂)</td><td rowspan="4">Bit in Magnitude of (m₂) determines number of words printed</td></tr><tr><td>Addr.</td><td>Addr.</td><td>is mode</td></tr><tr><td>00</td><td>02</td><td>Octal</td></tr><tr><td>01</td><td>03</td><td>Decimal</td></tr><tr><td>10</td><td>-</td><td>Alpha-betic</td></tr></table>	No	I of (m ₂)	Bit in Magnitude of (m ₂) determines number of words printed	Addr.	Addr.	is mode	00	02	Octal	01	03	Decimal	10	-	Alpha-betic
No	I of (m ₂)	Bit in Magnitude of (m ₂) determines number of words printed																		
Addr.	Addr.		is mode																	
00	02		Octal																	
01	03		Decimal																	
10	-	Alpha-betic																		
Halt	ht - 22	2100 or 3000	2100 or 3000	Fill from Flex.	Computer to rest; wait for Flexowriter															
Fill	fl - 11	2100 or 3000	2100 or 3000	Fill from Flex.	Computer to rest; fills from Flexowriter															
Buffer Out	bo - 04	2100 or 3000	2100 or 3000	{ First of 8 Memory Cells }	{ Least significant digit of m ₃ corresponds to least significant digit of first buffer cell }															
Buffer Load	bl - 05	2100 or 3000	2100 or 3000																	
Block Search	bs - 14	2100 or 3000	- T - A	A A A A	{ T: tape unit A: block address (0 0000 thru 1 7777) K: tape section; buffer cell }															
Write Tape	wt - 15	2100 or 3000	- T - K	- - - O																
Read Tape	rt - 16	2100 or 3000	- T - K	- - - N																
Read Card	rc - 06	Input	Addend	- - - N	{ N: first of the four buffer cells involved in the command }															
Punch Decimal	pd - 12	Input	Addend	- - - N																
Punch Octal	po - 13	Input	Addend	- - - N																

TABLE III-4. Command List

quency since the drum is driven by a synchronous motor. However, as all logical operations in the computer are synchronized to this clock waveform, minor variations in drum rotation speed ($\pm 5\%$) do not affect the process of computation.

A synchronizing pulse, obtained from the word channel, is used for synchronizing the pulse and octal digit counters.

COMMAND CODE

TABLE III-4 lists the commands used in the 102-A. As previously mentioned, octal digits 0_{13} and 0_{12} of a command are the Operation digits, and they may appear in any octal combination from 00 to 37.

Arithmetic Commands

The arithmetic commands are used to perform the bulk of all computation, to modify commands, and for all other purposes which the programmer might assign to them. In general, most functions of the computer involve the combined use of many commands; therefore, it is recommended that the reader acquaint himself with all of the commands.

In certain functions, commands are operated on as if they were numbers. In such cases the two octal digits which identify the command are treated as sign digits, except in the ADD and SUBTRACT commands. The following information is given for each command.

1. Name, abbreviation, and octal code digits.

2. Minimum execution time. The final execution time would depend on the location of the commands on the surface of the drum which in turn depends on the word channel configuration. The quoted figure assumes optimum placement of the operands.

3. Description of the function of the command.

4. Examples where required.

A. 1. ADD (ad) 35
2. 7.4 or 8.2 milliseconds, depending on the sign of the addends.

3. a. If (m_1) is an octal number, add (m_1) to (m_2) with due regard to the algebraic sign of each, and record the sum in cell m_3 . If the magnitude of the sum requires more than thirty-six binary digits to be represented in the machine, record an overflow digit in the overflow position of the sign (see example). When an overflow digit is so generated, the computer will cause an overflow alarm unless the next command is TEST FOR OVERFLOW MARKER or SHIFT LOGICALLY. (The generation of an overflow alarm may be suppressed by placing the console switch marked AUTOMATIC OVERFLOW TEST, in the OUT position.) If an overflow marker existed in the (m_1) before the add, an overflow marker will record in m_3 , but this will not cause the overflow alarm unless the sum of the (m_1) and (m_2) exceeds thirty-six binary digits. An overflow marker in (m_2) has no effect on the sum.

b. If (m_1) is a command, add the absolute value of (m_1) to the algebraic value of (m_2) and record the sum in m_3 . The two octal digits identifying the command in (m_1) are retained in m_3 . Any overflow generated in taking the sum will not cause the overflow alarm.

4. Examples: (Decimal numbers used in numerical examples.)

(m_1) +500000000

(m_2) +700000000

(m_3) p200000000

Overflow generated

(m_1) p314159264 (m_1) +271828183

(m_2) +271828183 (m_2) p314159265

(m_3) p585987448 (m_3) +585987448
No overflow generated

(Octal Commands)

 (m_1) 36 0364 0277 1630 (m_2) 00 0001 0010 0054 (m_3) 36 0365 0307 1704 (m_1) 00 0001 0010 0054 (m_2) 36 0364 0277 1630 (m_3) 02 0363 0267 1554 (m_1) 36 0364 0277 1630 (m_2) 00 7413 7500 6160 (m_3) 36 0000 0000 0010

No overflow generated

B. 1. SUBTRACT (su) 36

2. 7.4 or 8.2 milliseconds, depending on the sign of the operands.

3. The SUBTRACT operation in the 102-A is exactly like the ADD operation with the exception that the algebraic sign of (m_2) is inverted before the addition takes place.

C. 1. MULTIPLY AND ROUND-OFF (mr) 25

2. 25.0 milliseconds.

3. Multiply the algebraic values of (m_1) and (m_2) as if the binary points were at the left end of the magnitude, i. e., as if (m_1) and (m_2) both were binary fractions, and record the most significant half of the product rounded off to thirty-six binary digits (twelve octal digits) with the proper algebraic sign in cell m_3 . A product is always a positive or negative number without overflow. Any overflow marker in (m_1) or (m_2) has no effect on the product.

D. 1. MULTIPLY DOUBLE LENGTH (md) 26

2. 25.0 milliseconds.

3. Multiply the algebraic values of (m_1) and (m_2) and record the least significant thirty-six binary digits (twelve octal digits) of the product with proper algebraic sign in cell m_3 and record the most significant thirty-six binary digits (twelve octal digits) of the product with proper algebraic sign in the next cell adjacent to m_3 on the surface of the drum. A product is always a positive or negative number without overflow. Any overflow marker in (m_1) or (m_2) has no effect on either half of the product.

E. 1. DIVIDE AND ROUND-OFF (dr) 23

2. 25.8 milliseconds.

3. a. Divide the algebraic value of (m_1) by the algebraic value of (m_2) , treating (m_2) as if the binary point were at the left end of the magnitude, i. e., as if (m_2) were a fraction, and record the quotient rounded off to thirty-six binary digits with the proper algebraic sign in m_3 . Note that the quotient will always be greater than the dividend.

b. The divide command will generate an overflow digit if the absolute dividend is greater than or equal to the absolute divisor. This function has no regard for any arbitrarily assigned scale factors; for this purpose, the machine may be assumed to consider both divisor and dividend as if the binary point were at the left end of the magnitude. If, in such a case, the absolute quotient is less than two, the quotient recorded in m_3 is correct with the unit appearing in the overflow digit position. If, in such a case, the absolute quotient is equal to or greater than two, the result recorded in m_3 is meaningless. In any case, when an overflow digit is generated, the computer will cause an overflow alarm, un-

less the next command is TEST FOR OVERFLOW or SHIFT LOGICALLY. This feature may be suppressed by placing the console switch marked AUTOMATIC OVERFLOW TEST in the OUT position.

F. 1. DIVIDE AND SAVE REMINDER (dd) 24

2. 25.8 milliseconds.

3. a. Divide the algebraic value of (m_1) by the algebraic value of (m_2) , treating (m_2) as if the binary point were at the left end of the magnitude, i. e., as if (m_2) were a fraction, and record the remainder with the same sign as (m_1) in m_3 and the first thirty-six binary digits of the quotient with the proper algebraic sign in the next cell adjacent to m_3 on the surface of the drum (see examples).

b. The overflow feature of DIVIDE AND SAVE REMAINDER is the same as for the DIVIDE AND ROUND-OFF command.

4. Examples: (Decimal numbers used in example)

(m_1)	-100000000
(m_2)	+300000000
Remainder in m_3	-100000000
Quotient in adjacent cell	-333333333
(m_1)	+000000375
(m_2)	+000000200
Remainder in m_3	+000000000
Quotient in adjacent cell	p875000000

G. 1. SHIFT MAGNITUDE (sm) 30

2. 8.6 milliseconds, basic time plus 0.78 milliseconds for each shift.

3. a. Shift the magnitude of (m_1) according to (m_2) and record the shifted value in m_3 . Digits shifted off to the left or right end of the magnitude are lost and zeros are shifted in on the opposite end.

b. (m_2) is interpreted as follows: If (m_2) is positive, shift left. If

(m_2) is negative, shift right. The number of binary shifts is the entire magnitude of (m_2) treated as an octal integer.

4. Examples: Effect of sm m_1 m_2 m_3 on the following numbers.

m_1 (octal command)	26 0143 1163 2777
m_2 (shift control)	+ 0000 0000 0030
m_3 (result)	26 2777 0000 0000

H. 1. SHIFT LOGICALLY (sl) 27

2. 9.0 milliseconds, basic time plus 0.78 milliseconds for each shift.

3. a. Shift (m_1) including the sign digits according to (m_2) record the shifted value in m_3 . Digits shifted off the left or right end of the word are lost, and zeros are shifted in on the opposite end. In this command, it must be noted that the leftmost binary digit of the word is always a zero and therefore the leftmost octal digit may not exceed three (see example).

b. (m_2) is interpreted in SHIFT LOGICALLY exactly as in SHIFT MAGNITUDE.

4. Example: Effect of sl m_1 m_2 m_3 on the following numbers.

m_1 (octal command)	26 0143 1163 2777
m_2 (shift control)	+ 0000 0000 0030
m_3 (result)	23 2777 0000 0000

Note that the octal number six has been recorded as octal number two because of the restriction of the binary digit at the left end of the word.

I. 1. SCALE FACTOR (sf) 31

2. 10.6 milliseconds basic time plus 0.78 milliseconds for each shift.

3. a. If (m_2) has anything in the magnitude section, shift only the magnitude left until a binary one appears in the most significant binary digit of the magnitude. If (m_2) is zero, do not shift.

b. Subtract the number of required shifts from (m_1) , treated as an

octal integrator. If (m_2) is zero, subtract sixty-four (octal 100) from (m_1) , treated as an octal integer.

c. Record the modified (m_1) in cell m_3 and the shifted results in the next cell adjacent to m_3 on the surface of the drum.

4. Example: Effect of sf m_1 m_2 m_3 on the following numbers.

m_1 (octal scale factor) +0000 0000 0030
 m_2 (octal command) 26 0143 1163 2777
 m_3 (resulting octal scale factor) +0000 0000 0023
 Next adjacent cell
 (resulting command) 26 6144 7153 7740

Logical and Transfer Commands

A. 1. EXTRACT (ex) 32

2. 32.4 milliseconds if m_3 is in main memory; 10.6 milliseconds if m_3 is in buffer register.

3. a. The extract command moves selected binary digits from one word into the same positions of another word. From (m_1) , those binary digits which are in the same positions as the binary ones of (m_2) are inserted into the corresponding positions of m_3 . Where there are binary zeros in (m_2) the corresponding positions of (m_3) remain unchanged.

4. Examples:

(m_1) ad 0107 1523 2007
 (m_2) 00 0000 7777 0000
 (m_3) mr 2007 0674 2000
 Resulting (m_3)
 mr 2007 1523 2000
 (m_1) 10 7236 3521 4263
 (m_2) 30 2362 4125 3637
 $m_3 = 2103 = \text{cell } 2100$
 therefore $(m_3) =$
 00 0000 0000 0000
 Result in 2003
 10 2222 0121 0223

B. 1. TEST FOR OVERFLOW MARKER (to) 37

2. 6.6 milliseconds

3. Examine (m_1) as if it were an octal number. If the sign contains a binary one in the overflow digit, take the next command from cell m_3 ; otherwise, proceed normally. A binary one may appear in the overflow digit for any one of the following reasons:

a. It may be filled initially as a word with an overflow marker.

b. It may be a command noted as having a one in the overflow digit.

c. It may be the result of an ADD, SUBTRACT, DIVIDE, or DIVIDE AND SAVE REMAINDER command which generated an overflow.

d. It may have been placed there by the SHIFT LOGICALLY command.

C. 1. TEST MAGNITUDE (tm) 34

2. 7.8 milliseconds when test works.

3. a. Compare the absolute magnitudes of (m_1) and (m_2) . If $|(m_1)| > |(m_2)|$ take the next command from cell m_3 ; otherwise proceed normally.

D. 1. TEST ALGEBRAICALLY (ta) 33

2. 8.2 milliseconds.

3. Compare (m_1) and (m_2) algebraically. If $(m_1) > (m_2)$ take the next command from cell m_3 ; otherwise, proceed normally. The TEST ALGEBRAICALLY command considers positive zero greater than negative zero.

E. 1. TEST SWITCH OR TEST SEARCH (ts) 17

2. 6.3 milliseconds.

3. a. If m_1 is 2010, 2020, 2040, or 2100 and the respectively numbered switch on the console is in the up position, take the next command from cell m_3 ; if the switch is down, proceed normally.

b. If m_1 is 2200 and any Magnetic Tape Unit attached to the computer is block searching, take the next command from cell m_3 . If no Magnetic Tape Unit is searching, proceed normally.

F. 1. HALT (ht) 22
2. 9.3 milliseconds.

3. a. Stop computation and return machine to idle.

b. Prepare computer to accept octal.

c. Leave control register prepared to place the first word entered from the Flexowriter by the normal filling operation into cell m_3 .

G. 1. BUFFER OUT (bo) 04
2. 10.2 milliseconds.

3. Transfer the entire eight words of the buffer register into the eight physically adjacent main memory cells starting with m_3 . The word which will be entered into m_3 is the one whose buffer cell has the same least significant address digit as the address m_3 . The BUFFER OUT command does not clear the Buffer Register; it merely copies the contents of the latter into the Main Memory.

H. 1. BUFFER LOAD (bl) 05
2. 9.8 milliseconds.

3. Transfer the eight words in the physically adjacent main memory cells, starting with m_3 , into the buffer register. The word in cell m_3 will be entered into that cell of the buffer register which has the same least significant address digit as the address m_3 . The BUFFER LOAD command replaces the

previous contents of the Buffer Register with the contents of eight adjacent Main Memory cells as described; the previous contents are lost. The Buffer Register may be cleared with the command bl 3000 3000 2100.

Input-Output Equipment Commands

Flexowriter Commands:

A. 1. PRINT (pr) 21

2. Approximately 100 milliseconds are required for each character printed.

3. a. The total number of words to be typed out with the one PRINT command is m_3 taken as an octal integer; m_3 must be greater than zero.

b. The first word to be typed is (m_1). If more than one word is typed, the following words are ($m_1 + 1$), ($m_1 + 2$)... ($m_1 + m_3 = 1$). All words are typed in the same mode as the first word.

c. The printing mode: i.e., octal printing, decimal printing, or alphabetic printing is controlled by the sign digits of (m_2) according to the following table.

m_2 sign	Mode of Printing
00	octal
01	decimal
02	octal with address
03	decimal with address
10	alphabetic

d. The number of digits which will be printed from the magnitude section of each word is controlled by the magnitude section of (m_2). This is effected by placing in (m_2) a binary one in the least significant binary digit position of the octal, decimal, or alphabetic digit position corresponding to the last digit which is to be typed from a single word. This feature affects only the magnitude section of all three printing modes.

4. Examples:

pr 1726 2100 0002
 (1726) = 35 2000 2001 2003
 (1727) = 26 2003 1325 2003
 (2100) = 00 0000 0000 0000
 result: 35 2000 2001 2003
 26 2003 1325 2003

pr 1001 0204 0001
 (1001) = 14 1761 2025 4400
 (0204) = 03 0000 0000 0400

result: 1001 3.14159

pr 0000 0133 0010
 (0000) = 00 3675 3274 4564
 (0001) = 00 3666 3241 7560
 (0002) = 00 7766 4171 6436
 (0003) = 00 6232 4144 7464
 (0004) = 00 3663 3245 7360
 (0005) = 00 4475 4563 6436
 (0006) = 00 6232 7776 7041
 (0007) = 00 6654 1212 1234
 (0133) = 10 0000 0000 0000

result: The National Cash Register Company

B. 1. FILL (fl) 11

2. Indeterminate.

3. a. Stop computation and return machine to idle.

b. Prepare computer to accept octal.

c. Leave Control Register prepared to place the first word entered from the Flexowriter into cell m_3 in the normal filling operation.

d. Start Flexowriter Paper Tape Reader.

e. When the read-in process is finished, and if computer is not restarted with an "s" code punched in the tape, the computer will remain in the idle condition.

IBM Equipment Commands:

A. 1. PUNCH OCTAL (po) 13

2. Approximately 550 milliseconds.

3. a. The first of the four Buffer Register cells that are to be punched is specified by the least significant digit of m_3 . The first cell is the one whose address has the same least significant digit as m_3 .

b. The contents of the Comparison Register are specified by (m_1). For the normal octal punching process this value is as follows:

2	6	6	6	6	6	6	6	6	6	6	6	6	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---

c. The contents of the Addend Register are specified by (m_2). For the normal octal punching process this value is as follows:

1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---

B. 1. PUNCH DECIMAL (pd) 12

2. Approximately 650 milliseconds.

3. a. The first cell of the Buffer Register of the four whose contents are to be punched is specified by the least significant digit of m_3 . The first cell is the one whose address has the same least significant digit as m_3 .

b. The contents of the Comparison Register are specified by (m_1). For punching an X in a particular column when the negative sign is coded as a two, this value is:

0	2	7	3	5	6	7	3	5	6	7	3	5	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---

For punching numerical data in all ten columns the value is:

1	6	7	3	5	6	7	3	5	6	7	3	5	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---

c. The contents of the Addend Register are specified by (m_2). For punching the sign digit in the X row this value is:

0	0	0	4	2	1	0	4	2	1	0	4	2	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---

For punching numerical data in all ten columns this value is:

0	1	0	4	2	1	0	4	2	1	0	4	2	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---

C. 1. READ CARD (rc) 06

2. Approximately 550 milliseconds for octal reading, 650 milliseconds for decimal reading.

3. a. The first of the four Buffer Register cells into which data is to be entered is specified by the least significant digit of m_3 . The first cell is the one whose address has the same least significant digit as m_3 . Thus, if m_3 is 0000, the first cell would be 2000 and the following three would be 2001, 2002, and 2003. If m_3 is 1726, the first cell would be 2006 and the following three would be 2007, 2000, and 2001.

b. The contents of the input register are specified by (m_1) . This is a different value for the normal octal reading process than for the normal decimal reading process.

(m_1) when reading octal numbers or commands:

1	5	5	5	5	5	5	5	5	5	5	5	5	6
---	---	---	---	---	---	---	---	---	---	---	---	---	---

(m_1) when reading decimal numbers with sign digit used in X punch:

0	1	6	7	3	5	6	7	3	5	6	7	3	6
---	---	---	---	---	---	---	---	---	---	---	---	---	---

(m_1) when reading decimal numbers with sign digit used for numerical data:

1	5	6	7	3	5	6	7	3	5	6	7	3	6
---	---	---	---	---	---	---	---	---	---	---	---	---	---

c. The contents of the Addend Register is specified by (m_2) . This is a different value for the normal octal reading process than for the normal decimal reading process.

(m_2) when reading octal numbers or commands:

1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---

(m_2) when reading decimal numbers:

0	1	0	4	2	1	0	4	2	1	0	4	2	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---

Magnetic Tape Unit:

- A. 1. BLOCK SEARCH (bs) 14
2. Less than 20 milliseconds computer time; indeterminate tape unit time.

3. a. A BLOCK SEARCH command has the following configuration:

b	s					A_c			A_t		
---	---	--	--	--	--	-------	--	--	-------	--	--

b. A_c is a single octal digit which specifies the tape unit in which the search is to take place. In systems with only one tape unit, A_c is always zero and may be omitted.

c. A_t is five octal digits which specify the address on the tape which is to be searched for.

- B. 1. READ TAPE (rt) 16
2. Approximately 170 milliseconds.

3. a. A READ TAPE command had the following configuration:

r	t					A_c	K				N
---	---	--	--	--	--	-------	---	--	--	--	---

b. A_c is a single octal digit which specifies which tape unit will be read from. In systems with only one tape unit, A_c will be zero and may be omitted. If the tape unit specified by A_c is block searching when the READ TAPE command is given, reading will not commence until the block search process is complete. Only eight words of one block are read with one READ TAPE command. In order to be read, the tape must move; therefore, after every READ TAPE command on a specific tape unit, the next READ TAPE command before a BLOCK SEARCH command on that specific tape unit will cause the next adjacent block to be read.

c. The octal digit K determines the reading mode as follows:

If K is equal to zero, read the Block Address word into the Buffer Register cell whose address has the least significant octal digit of N.

If K is equal to one, read the odd words of the block into the Buffer Register. The first word will read into the Buffer

Register cell whose address has the least significant octal digit of N . The remaining seven words will read into the Buffer Register in cells $200N + 1$, $200N + 2$, and so forth. When a word has been read into cell 2007, the next word, if there is one, will be read into cell 2000, since Buffer Register operation is cyclic.

If K is equal to two, read the even words of the block specified by A_t into the Buffer Register in the same fashion as described for the odd words in the above paragraph.

- C. 1. WRITE TAPE (wt) 15
 2. Approximately 170 milliseconds.
 3. a. A WRITE TAPE command has the following configuration:

w	t					A_c		K				O
---	---	--	--	--	--	-------	--	---	--	--	--	---

b. A_c is a single octal digit that specifies which tape unit will be written on. In systems with only one tape unit, A_c will be zero and may be omitted. If the tape unit specified by A_c is block searching when the WRITE

TAPE command is given, writing will not commence until the Block Search process is complete. Only eight words of one block are written with any one WRITE TAPE command. In order to be written on, the tape must move; therefore, after every WRITE TAPE command to a specific tape unit, the next WRITE TAPE command before a BLOCK SEARCH command to that specific tape unit will cause the next adjacent block to be written on.

c. The octal digit K determines the writing mode as follows:

If K is equal to zero, write the word in Buffer Register cell 2000 into the Block Marker word of the selected block.

If K is equal to one, write the contents of the Buffer Register into the odd words of the selected block in the following manner: Cell 2000 to word No. 1, cell 2001 to word No. 3, and so on until cell 2007 has been written in word No. 15.

If K is equal to two, write the contents of the Buffer Register into the even words of the selected tape block in the same fashion as it was written in the odd words.

Theory

INTRODUCTION

The theory of operation of the Model 102-A General Purpose Digital Computer is based upon two-valued symbolic logic. This logic is handled in mathematical form according to a set of rules called Boolean algebra, and once established, as shown by the logical equations in Volume II, represents the logical circuitry within the computer. The outputs from these logical circuits can be one of two values; +125 volts to represent a true state of a proposition or a binary one, or +100 volts to represent a false state of a proposition or a binary zero.

By defining certain logical operations, propositions can be combined in various ways to achieve certain desired results in the computer. The operations used are "and" (logical product, gating, or coincidence), "or" (logical sum or mixing), and negation ("prime", "not", or inversion). The basic electronic elements used to perform these operations are vacuum tube flip-flops, diode networks, drivers or inverter tubes, and cathode followers.

In addition to the logical circuitry, the computer has its own internal memory system. This system consists of a memory drum, read-write heads and associated record and playback amplifiers, and memory flip-flops. The memory enables the computer to store and look-up information, and to supply the timing pulses (clock) to the logical circuits. These pulses are used to trigger the counters and various flip-flops within the logical networks so that all components operate in a predeter-

mined time-sequence pattern.

Because this time-sequential type of operation imposes a multiplicity of uses on the various circuits, conventional block diagrams cannot readily represent the sequence; and, therefore, a flow diagram is used to illustrate the operations within the computer during any given routine. For this reason, a thorough knowledge of the theory of operation and a complete understanding of the flow diagram and logical equations become indispensable for fast, effective, corrective maintenance.

The input-output equipment circuitry and power supply complete the computer configuration. Voltages from the power supply enable the computer to perform its various operations and the input-output circuitry together with the various input-output equipment enable the computer to communicate.

POWER SUPPLIES

The power required to start and run the Model 102-A computer, as indicated in TABLE I-2 (page I-5), is about 7.7 kva of 230-v, single phase, 60-cycle, alternating current. If the input power available is only 208v, a step-up transformer must be installed to bring the voltage up to 230v.

As shown in Figure 20 (1 of 3), Vol. II, this power is first applied through transformer T8 and a bridge rectifier RXI to obtain +85-v dc to operate the relays.

The sequencing of these relays is discussed in detail further on.

The principal d-c voltages used in the computer are obtained through transformers T5, T6, and T7. The primaries of both T5 and T7 are energized directly from the input source. T6, however, has its primaries connected to the output of the stabiline unit which provides + 1 percent regulation of the 230-v ac. The principal d-c voltages are listed in TABLE IV-1.

The regulated power supplies are of both the series (+225 and -300v) and shunt (+100v) types of quite conventional design. For example, in the +225 regulated supply which is obtained from transformer T5 through a selenium rectifier, see Figure 20 (2 of 3) Vol. II, the 6080 dual triode is used as a series voltage regulator, controlled by the 6AU6 difference amplifier (V2). Thus, if the +225v supply rises at its take-off point (terminal 8 of resistor R114 of test jack J4), a corresponding rise in voltage would appear at the grid of the 6AU6, whose cathode is held at constant voltage by the OB2 gas-filled VR tube (V3). As a result the plate of V2 falls, bringing down the grids of the 6080 and thus lowering the +225 to its proper value. This is indicated in Figure IV-1.

The -300v regulated supply is series regulated in a similar way, except that here the cathode of the VR tube (V11) is connected to -300v and its plate to the cathode of the 6AU6 difference amplifier. If the -300 goes to -301v, for example, this lowers the grid of V10 and its plate goes up, bringing up the grids of the series regulator tube, V9. As a result the tube conducts more; and the voltage rises in the circuit of R52, R51, and R50, bringing the negative supply back to -300v.

The +100v regulated supply is shunt regulated, utilizing two 6080 tubes (V5 and V6) between the +100v and ground, as well as a difference amplifier and VR

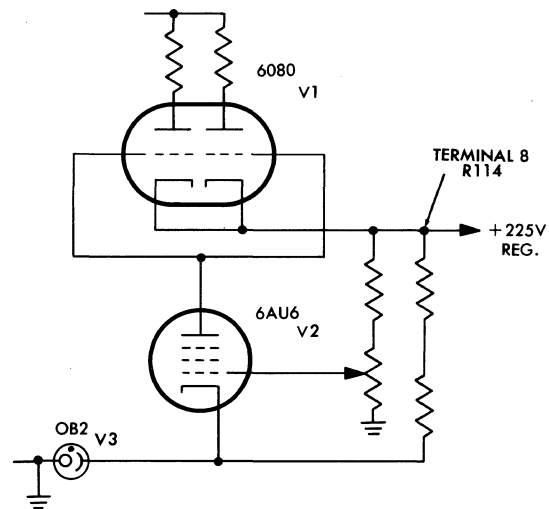


Figure IV-1. Voltage Regulation in the 225-V Supply

tube as before. If the +100v attempts to rise, this rise brings up the cathode of the OA2 (V8) and hence the cathode of the 6AU6 (V7). Since the grid rise on V7 is less than the cathode rise, its plate will go up, bringing up the grids of the 6080 tubes. Since their cathodes are tied to ground, these tubes will conduct more, bringing down the +100 to its proper value.

CLAMP AND BIAS VOLTAGES

The +180/+170v, required for upper and lower clamping diodes of the memory limiter amplifier input circuit, is developed from 225v (regulated) in a dividing circuit located on the memory clamp chassis. This voltage is regulated by the R4 potentiometer (see Figure 14, Vol. II). The +150 for the upper clamping diode of the record tube driver comes from tap 2 of R64, a bleeder resistor connected to the +225v (unregulated) source. The +125 comes from tap 5 of the same resistor, and +115 for the record tube driver's lower clamping diode from the lower end (terminal 38) of dropping resistor R91 from the +125.

D-C VOLTAGES	SCHEMATIC in VOL. II	APPLICATION IN COMPUTER
+415	Fig. 20 (2 of 3)	Plates of memory record tubes
+225 to 250 (regulated)	Fig. 20 (2 of 3)	Plates of main memory playback amplifiers.
+225 (unregulated)	Fig. 20 (3 of 3)	Plates of most other tubes; logical product resistors. Source of nu- merous other voltages, including 150, 125, 100, and 115v.
+180 } +170 }	Fig. 14	Clamping diodes of memory lim- iter amplifier input. Obtained from +225v regulated.
+150	Fig. 20 (3 of 3)	Clamping diodes of record tube driver.
+115	Fig. 20 (2 of 3)	Clamping diodes of record tube driver, obtained from +125v.
+125	Fig. 20 (3 of 3)	Upper clamping diodes; true log- ical voltage level for logical sums and products; recirculating line record tube cathodes.
+100	Fig. 20 (3 of 3)	Lower clamping diodes (clamped flip-flops, flip-flop drivers, logi- cal drivers, clamped one-shots, clippers); false logical voltage level for logical sums and products; screen grids of record tube driver; recirculating line record tube cath- odes.
+20	Fig. 20 (2 of 3)	Upper clamp bias and lower clamp bias for the memory clipper. Ob- tained from +100v regulated.
+10		
+6	Fig. 20 (2 of 3)	Noise clipper bias, obtained from +100v regulated.
-6	Fig. 20 (2 of 3)	Grid clamp bias, obtained from -300 regulated.
-10	Fig. 20 (2 of 3)	Thyratron grid bias, obtained from -300 regulated.
-300	Fig. 20 (3 of 3)	Grids or cathodes of numerous tubes; source of negative bias volt- ages.

TABLE IV-1. Principle D-C Voltages

The +20 for the memory clipper's upper clamp bias is obtained from the +100v (regulated) at the lower end of R30 (terminal R). The +10 for the clipper's lower clamping bias comes from the same supply at terminal S between R80 and R81. The noise clipper bias of +6v comes from tap T on potentiometer R21, also connected (through R98) to the +100v supply.

The -6v for grid clamp bias is obtained from potentiometer R31, connected through resistors R99 through R102 to the -300v supply and through R105 to ground. The -10v used for thyatron grid bias for the 57 thyatrons required for IBM card punching, is also obtained from the -300v supply at terminal M near the upper end of resistor R104.

FILAMENT BIAS FOR REGULATOR TUBES

Note that large voltages are applied to the cathodes of the series and shunt regulator tubes. To eliminate voltage breakdowns between the filaments and cathodes, the outputs of the regulated power supplies are fed back to the appropriate filament transformers. Thus the +225 is applied at the center tap of T1, filament transformer for the 6080 dual triode used in series-regulating the +225 supply. The +100 is applied to T3, filament transformer for V4 (the 5963 in the comparator circuit discussed below) and for V2 (the 6AU6 difference amplifier in the +225 regulation circuit). The -300v is applied to T4, filament transformer for V7 (the 6AU6 in the +100v regulated supply) and for V10 (the 6AU6 in the -300 supply).

COMPARATOR CIRCUIT

Instead of regulating the +125v supply, a comparator circuit is provided to compare the difference between the +125v and the regulated +100v, and to turn off the d-c voltages supplied to the logical networks if this difference is

excessive, see Figure 20 (2 of 3), Vol. II. This safeguards the many diodes in the logical networks from prolonged exposure to excessive back voltages which could be damaging.

In the comparator circuit, a fixed opposing bias voltage from a 45v dry cell battery is placed in series with the +125v and applied to the grid of one half of the 5963 dual triode comparator tube (V4) through R86 and R115. The resultant level at this grid is about +80v. The cathode of this half of the tube is tied to the +100v regulated, and its plate through R32 to the +225v regulated. Thus, normally, this left triode is cut off with negative grid bias. If the +125v supply rises, or the +100v drops, this grid will become less negative with respect to the cathode; and the plate voltage will drop.

Normally the right half of the comparator triode is turned on, with its grid biased positively because of the voltage divider action between the left plate connected through R83 to the grid, and then through R82 to -300v. However, when the left triode conducts and its plate drops, this brings down the right grid.

If this action should bring the grid down near cut off, comparator sensing relay K9 opens, which in turn opens the contacts of relay K12, removing the load from the d-c power supplies. If the differential voltage between the logical voltages returns to normal (25v), then relay K12 is automatically closed again after K9 closes. Thus the functioning of the comparator circuit is essentially that of a sensitive electronic circuit breaker.

RELAY SEQUENCING

This relay operational sequence is shown in block diagram form in Figure IV-2, and described briefly below. As mentioned previously, the 85-v dc required to operate the relays is obtained

from the external 230-v ac through transformer T8 and a bridge rectifier.

When the POWER ON button on the computer control panel is pressed, the start relay (K1) is energized. Since the POWER ON button is a momentary contact type, the self-holding contacts are closed when the voltage is first applied, thus keeping K1 energized. The contacts in series with the motor-driven time delay relay switch (K10) are closed, making this relay operative. The contacts in series with the main a-c contactor (K18) are closed and thus the a-c external power is applied to the primary power circuits, transformers T1 through T4, T5 and T7, and through the stabilizer to T6. When energized, K18 also places self-holding contacts and a current limiting resistor in series with itself.

Operating the POWER ON button also energizes the drum motor contactor relay (K20) so that the memory drum starts revolving.

K4, K5, K8 and K9 are single-gang, single-pole, single throw-relays which form a series link from the 85-v d-c supply either through the DC ON but-

ton on the computer test panel or through relay K10 in the normally closed position. The series link formed by these four relays acts as a voltage buildup sensing device and, as shown in Figure 20 (1 of 2), Vol II, may be jumpered during testing or servicing.

K4 is made operative by the +225-v regulated power supply, while K5 and K8 are the sensing relays of the +100 and -300v regulated supplies, respectively.

As noted in the discussion of the comparator circuit, K9 is the comparator sensing relay. It is energized and its contacts are closed by the +225v regulated supply. It remains operative so long as the potential difference between the +125v and +100v regulated is equal to or less than 25v.

When the voltage build-up has been completed in the regulated power supplies and the four sensing relays are closed, the d-c on-off control relay (K12) is energized through contacts 3 and 4 of K10 and the series of sensing relays. If, during servicing, it becomes necessary to close this circuit after K10 has opened, this may be accomplished by pushing the DC ON button on the computer

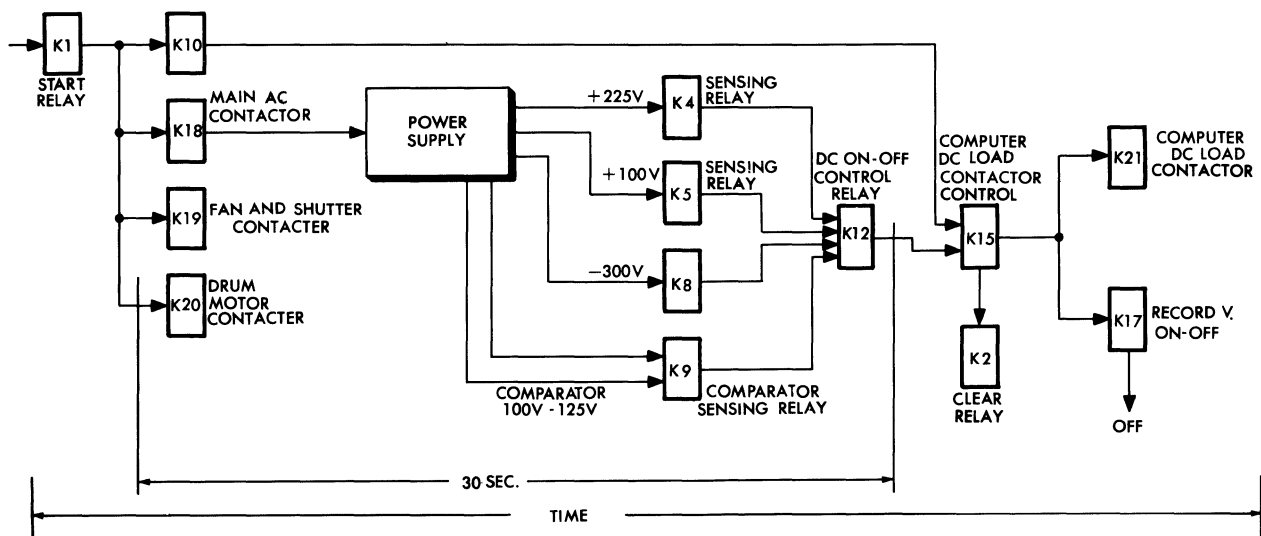


Figure IV-2. Relay Sequencing

test panel.

The d-c on-off control relay (K12) is a three-gang, single-pole, single-throw relay. Contacts 1 and 2 of K12 form a link between K10 and the computer d-c load contactor control, K15. Contacts 7 and 8 of K12 close, and with the energizing of K15 (contacts 7 and 8), provide a circuit for operating K17, the record voltage on-off relay.

K15, a three-gang relay, is energized when the contacts on the Haydon time delay relay (K10) close and K12 is also operative. Contacts 7 and 8 of K15 operate to remove the discharge resistor R1 (connected across capacitor C1) and apply voltage to K17. Note that the charging of C1 - see Figure 20 (1 of 3), Vol. II prevents the application of record voltage through K17 to the memory record circuits until the computer has been cleared. Contacts 1 and 2 of K15 operate to connect the computer d-c load contactor, K21, to the 85v supply. Contacts 4 and 5 of K15 close and parallel the K1 contacts in the circuit with K18, the main a-c contactor relay.

The record voltage on-off relay K17, a three-gang relay, is energized when contacts 7 and 8 of K12 and 7 and 8 of K15 are closed. Contacts 3 and 11 of this relay open and thus make K2, the clear relay, inoperative. Contacts 12 and 5 of K17 energize the WARM UP indicator light, while 12 and 4 energize the POWER ON indicator light. Both lights are mounted on the computer control panel. Contacts 13 and 6 operate to connect the +415v dc to the memory record circuits.

The clear relay, K2, remains energized until K17 is operative, or it may be made effective by pushing the CLEAR button on the computer control panel. As shown in Figure 20 (1 of 3), Vol. II, contacts 2 and 3 of K2 apply +125v to the plate of diode 27W on diode board

No. 4 when K2 is inoperative (proposition Z' true) and the clear proposition false. When K2 is operative, contacts 4 and 5 of this relay close and apply 125v wherever Z true appears in the circuitry.

The computer d-c load contactor, K21, is energized when K15 is on.

Self-holding contacts 2 and 3 of this relay close and insert a current limiting resistor (R74) in series with K21. When energized, K21 also opens its normally closed energizing contacts (T and F). Contacts 6 and 7 of K21 open and remove the dummy load from the +100v regulated supply, see Figure 20 (3 of 3), Vol. II. Contacts T3 and L3 close to apply the +100v to the computer load, while contacts T2 and L2 operate to apply the -300v to the computer load. The +150v is applied to the memory record tube driver by contacts T4 and L4, and contacts 8 and 9 close to apply the +125v to the computer load.

In the off sequence, that is, when the computer OFF button on the computer control panel is pressed, relay K1 opens and the other relays open in sequence as follows: K10, K19, K20, K15, K21, K17, K2, K18, K4, K5, K8, K9, and K12. The reason that K18 stays closed longer than K19 and K20 is the parallel contacts of K17, which hold K18 closed until K17 opens. This is done to keep voltage on the logical networks until the main memory record voltage (through K17) is removed. Also, after a few seconds the thermostat will cause the shutters to open and the fan to go on to keep the computer cool. This cycle may occur several times after the computer is turned off and before the a-c power is removed. Thus, by leaving the ac on for a brief period, the computer's internal temperature may be brought down to room temperature rapidly and automatically after shutdown.

LOGICAL CIRCUITS

The variety of logical sum and product

circuits that make up the diode network are formed by combinations of diodes and resistors (see Logical Circuit Development in the APPENDIX). These along with their associated drivers and flip-flops comprise the logical circuits that provide the electronic equivalent of the numerous logical equations describing the computer's operation. The function of the logical networks in the operation of the computer is indicated in Figure IV-3, which is a simplified block diagram of the logical structure of the computer.

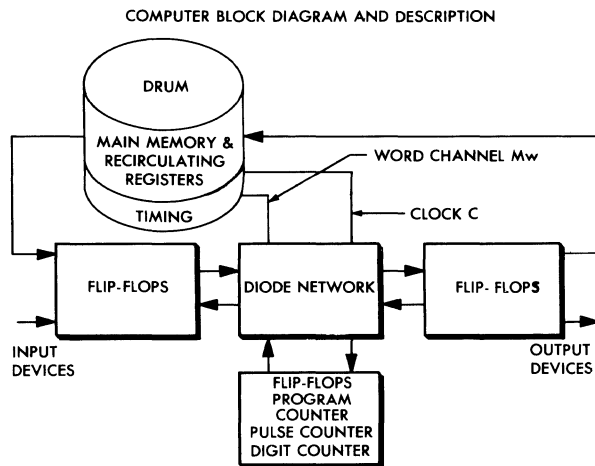


Figure IV-3. Logical Structure of the Computer

The propositions developed by the diode network are registered by the logical flip-flops. The program counter flip-flops control the interconnection of these logical elements. Thus, information enters the diode network from the memory and from input devices, and is fed out of the diode network back into the memory or into output devices. The clock on the memory drum provides the basic repetition rate for the computer.

FLIP-FLOPS, DRIVERS, CATHODE FOLLOWERS, ONE-SHOTS

In connection with the logical networks, the use of unclamped and clamped flip-

flops, flip-flop and logical drivers, cathode followers and one-shot circuits makes possible numerous operations in the computer.

The schematic of a typical unclamped flip-flop is shown in Figure IV-4. The flip-flop is a conventional Eccles-Jordan multivibrator with a specially designed input circuit. As indicated, a positive voltage of about +6v, called the noise clipper bias, is applied through R8. The effect of this bias is to make the flip-flop insensitive to positive pulses or negative noise pulses of small amplitude. Thus the flip-flop is triggered only by relatively large negative spikes which cut off the half of the tube that is conducting. When such a negative trigger pulse is applied, the voltage level on the grid side of diode D2 becomes less negative than cutoff and this half of the tube stops conducting. Its plate will rise sharply from about +100 to +200v, with this rise appearing at "False Output"; this brings up the grid of the other half of the tube and it starts conducting, the second stable state of the flip-flop.

Note that the grid of the right half of the tube receives a clamp bias of -10v from the cathode of diode D4. This is necessary because of the voltage divider effect of R1, R4, and R10 when the grid of the right half of the tube is cut off. Without the grid clamp bias, a voltage of -40 would appear on the grid. Since a rise of voltage on the opposite plate (connected to this grid through R4) tends to bring the grid voltage up to cutoff, the less negative the grid the more sensitive the flip-flop becomes.

The action of this unclamped flip-flop is further illustrated in Figure IV-5. Note that capacitor C4 and resistor R8 differentiate the input pulses. Thus a positive pulse coming in at the point marked "True Input" becomes a negative spike, plus a small positive rise at the leading edge.

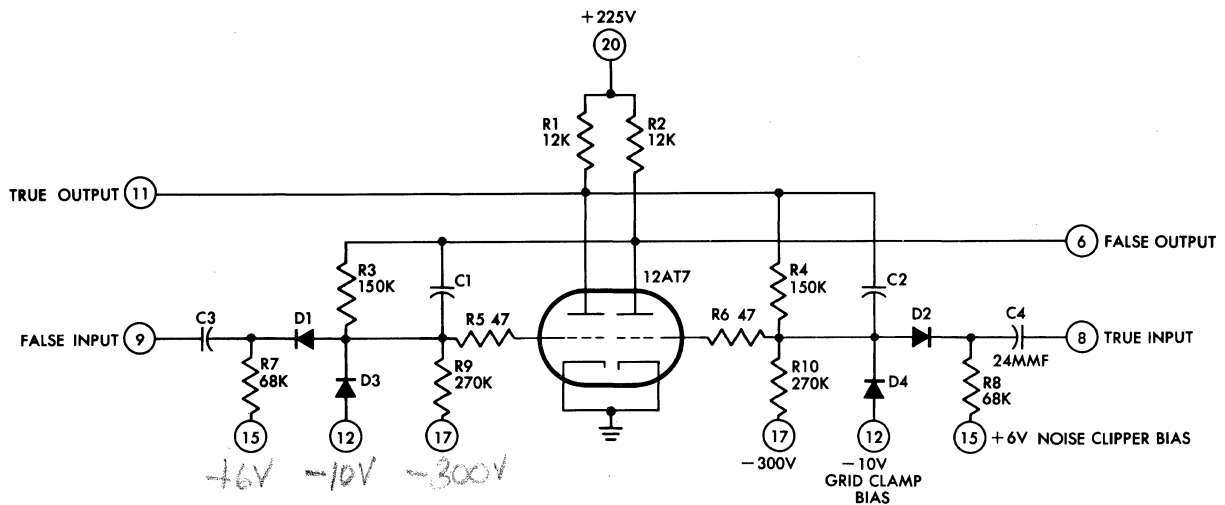
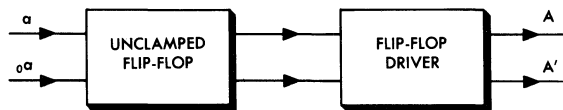


Figure IV-4. Typical Unclamped Flip-Flop

The outputs from such an unclamped flip-flop are connected to the inputs of a flip-flop driver (00367001) shown in Figure 21 (1 of 3), Vol. II. This driver is a dc-coupled triode amplifier. Note also that the driver outputs are diode clamped at the logical voltage levels of +125 and +100v. The following diagram illustrates that the combination of an unclamped flip-flop and a flip-flop driver produces a true output from the driver with the same polarity as the corresponding input to the flip-flop.



Another flip-flop driver design (00667-001) having a higher current capacity than the 00367001, is also shown in Figure 21 (1 of 3), Vol. II. Flip-flop driver (00567001), on the same schematic, has a lower current capacity than the 00367001 driver.

Similar drivers are used with inputs from a logical network or a clamped flip-flop. These are called logical drivers because the inputs are at the logical voltage levels instead of at +200 or

+100v as in the case of the unclamped flip-flops.

TABLE IV-2 summarizes the characteristics of various types of flip-flops and logical drivers.

The plate load resistor of a driver must have a small enough value to hold up the

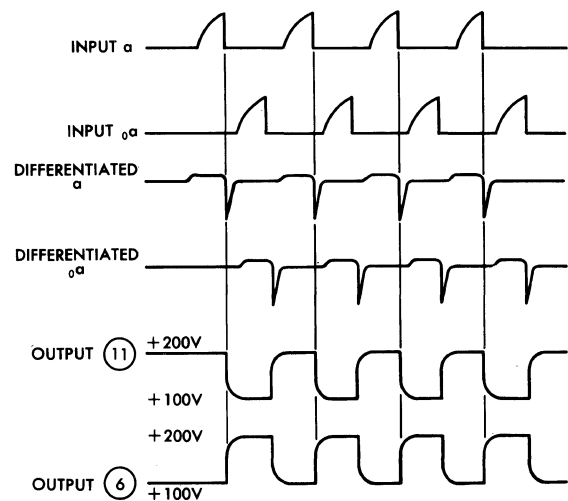
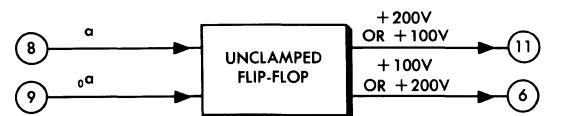


Figure IV-5 Idealized Waveforms For An Unclamped Flip-Flop

Schematic No.	Type	Tube Type Used	Approximate Current Capacity without Product Load
00267001	Logical Driver	5687 Dual Triode	45 ma.
00367001	Flip-Flop Driver	5687 Dual Triode	45 ma.
00467001	Logical Driver	5963 Dual Triode	12 ma.
00567001	Flip-Flop Driver	5963 Dual Triode	12 ma.
00667001	Flip-Flop Driver	5881 Pentode	51 ma.

TABLE IV-2. Flip-Flop and Logical Drivers

voltage of external sum loads to the proper level when the tube is not conducting. Also, the tube used must have enough current capacity to hold the plate resistor and external product loads down to the correct voltage level when the tube is conducting.

An exceptional use of the 5881 (006670-01) pentode driver is for amplifying the output from the 1.2 microsecond delay line, discussed on page IV-33. Here the input from the delay line is ac-coupled to terminal (7), and terminal (17) for grid bias voltage is tied in with terminal (4) at ground potential instead of being connected to the -300v supply, see Figure 21 (1 of 3) Vol. II.

Another driver circuit design, the double driver, is used to operate the logical relays. It uses two dual triode 6BQ7A's (12667102), as shown in Figure 21 (2 of 3), Vol. II, with a plate current per section of 13.5 ma.

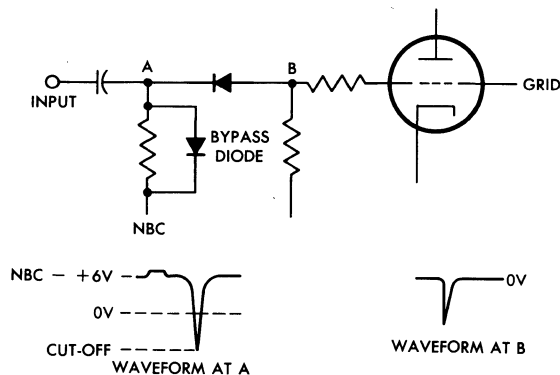
The use of diodes in clamp circuits, such as the outputs of the flip-flop and logical drivers, is to fix the voltage swing at the required logical voltage levels of +125 and +100v. The lower clamp diodes are tied to the 100v supply and fix the voltage swing at this level, while the upper clamp diodes are connected to the 125v supply and prevent the output from exceeding this value. Clamp diodes are used in sufficient

number to permit handling the same current loads as the driver can supply to an external load so that the clamping will be effective despite changes in this load. The polarity of the diodes is such that the voltage tends to drop below +100v or rise above +125v only by the amount of the forward IR drop in the diode.

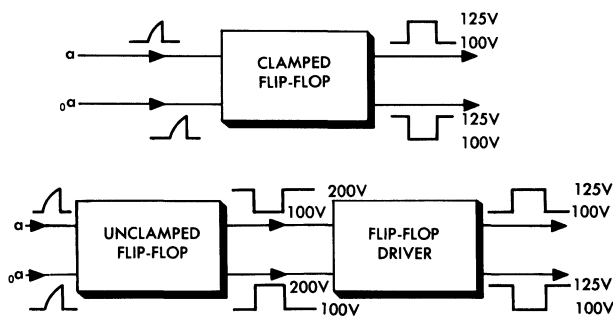
A typical clamped flip-flop (20267012) is shown in Figure 21 (3 of 3), Vol. II. Note that in this circuit there is no need for grid clamp bias at the inputs because the swing of the plate voltages is clamped at +100v for the conducting half-tube and +125v for the tube half that is turned off. Here voltage divider calculations show a grid voltage in the "off" half of about -14v, and at the other grid about +4v or less because of grid current.

Note also the by-pass diodes used with R7 and R8 in the input differentiating circuits. When the differentiated trigger pulse is of sufficient negative amplitude to cut off a conducting half-tube, the by-pass diode will appear as a large shunt impedance (the back resistance of the diode). The positive portion of a differentiated pulse, which is undesirable, is clamped by this by-pass diode to the noise clipper bias level. This is best illustrated by the waveforms shown

below of the differentiated input pulse at both sides of the input diode.



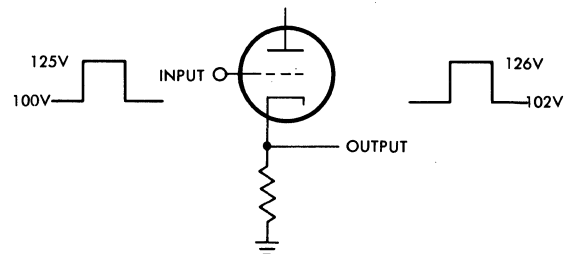
Note the difference in marking the outputs of the clamped flip-flop as compared with the unclamped version. A positive pulse at the true input of the unclamped flip-flop will, as has been shown, cut off this half of the tube and make the voltage level at the output marked true go down from about +200v to +100v. A similar pulse at the true input of the clamped flip-flop will make the true output (now the plate of tube that was conducting before the differentiated pulse hit it) go up from +100v to +125v. This difference occurs because an unclamped flip-flop is always followed by a flip-flop driver which inverts the signals so that the resultant true output corresponds to the true output from the clamped flip-flop. The diagrams below illustrate this point.



Another clamped flip-flop (20267002) with higher current capacity than the 20267012, using a 12BH7 double triode

instead of the 12AT7, is shown in Figure 21 (2 of 3), Vol. II.

The logical cathode follower (20267010), also shown in the same schematic, is used for decoupling a load from the input to avoid loading this source. As the following simplified schematic and waveform indicate, the input and output polarity are the same; however, there is a slight shift upward in the output level, since the cathode is at a slightly higher potential than the grid. The amplitude of the output is slightly lower than that of the input because the gain of the cathode follower is less than unity: This decrease in amplitude is about 5 percent.



The clamped one-shot (10667029), shown in Figure 21 (2 of 3), Vol. II, is used in the start circuit to trigger a clamped 12AT7 flip-flop. (See discussion of the start circuit, page IV-43.) A one-shot or univibrator has one stable state which can be triggered to the unstable state by a pulse of the correct amplitude and polarity. It will return to the stable state after a period depending upon the circuit constants. In this circuit, the period is proportional to the time constant C2R2. This is illustrated in Figure IV-6.

This one-shot has what is known as "dead time" while it is in its unstable state, because during this period it cannot be triggered. Thus a one-shot can be used to buffer the output from relays with contact bounce which might produce a train of undesired spurious pulses.

Clock Products and Delays

All the logical propositions in the com-

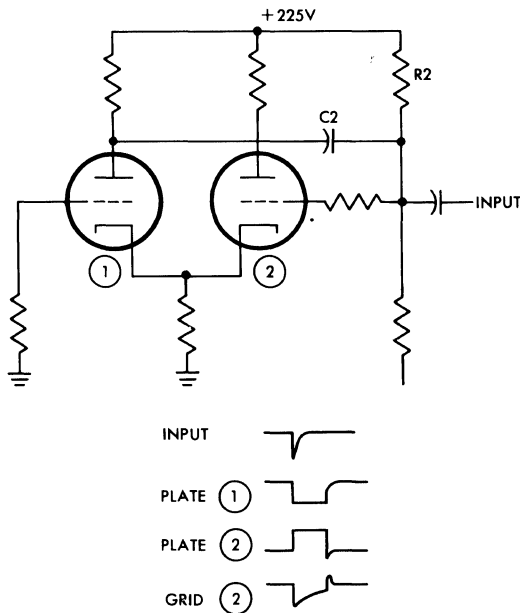


Figure IV-6. Typical Clamped One-Shot and Idealized Waveforms

puter which are used as input or grid propositions to trigger flip-flops are properly timed by introducing a clock pulse in such a way that the logical proposition and the clock pulse form a logical product (see Figure IV-7).

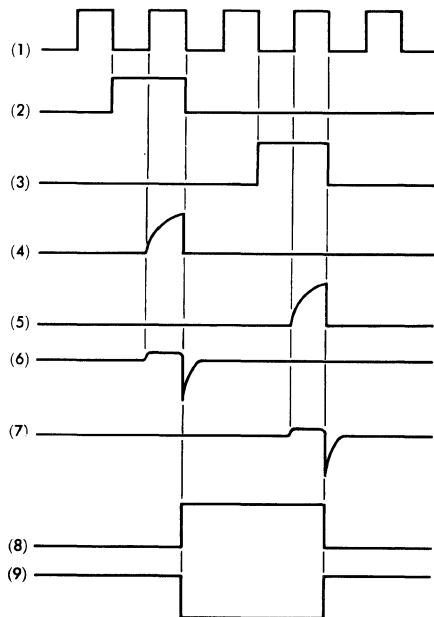
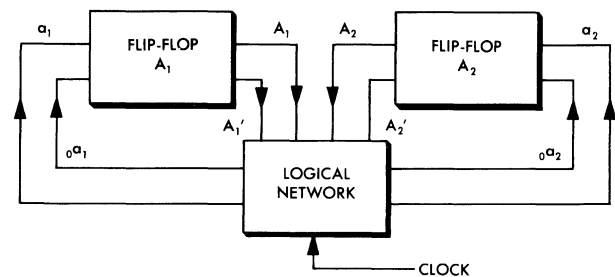


Figure IV-7. Idealized Waveforms of Grid Propositions

Note that the waveforms of QC and PC are not square waves because of the time constant of the clock product resistor and the circuit's stray capacitance. The trailing edge of this clock product becomes, when differentiated by the flip-flop's input circuit, the negative spike that triggers the flip-flop. Thus the flip-flop output is delayed by one pulse-time in relation to input.

Examples of Logical Design

To illustrate how diodes and resistors are utilized in the logical circuitry of sums and products, and a typical use for logical flip-flops, the design of a binary counter will be considered. The first step in designing a binary counter is to create the basic block diagram below, indicating two flip-flops, A_1 and A_2 , connected to a logical network and timed by pulses from the clock. C. This is a two-stage binary counter with A_1 the least significant binary digit.



The time sequence of the desired states of A_1 and A_2 can be tabulated as follows:

$t(\text{time})$	A_2	A_1
0_t	0	0
1_t	0	1
2_t	1	0
3_t	1	1

At this point the binary counter must be made to recycle. These desired states can be obtained only by varying the grid

signals, or inputs, a_1 , $o a_1$, a_2 , $o a_2$, to the flip-flops. Since each state of A_1 and A_2 determines the grid signals for the next state of A_1 and A_2 , the following truth table is developed.

Existing States		Required Grid Signals to get Next State				
t (time)	A_2	A_1	a_1	$o a_1$	a_2	$o a_2$
0_t	0	0	1	0	0	1
1_t	0	1	0	1	1	0
2_t	1	0	1	0	1	0
3_t	1	1	0	1	0	1

TABLE IV-3. Truth Table for Binary Counter.

Note that once a flip-flop is turned on, it stays in this stable state. Thus, it is unnecessary to make $o a_2$ true at 0_t or a_2 true at 2_t as indicated above.

A set of logical equations for the binary counter can be written from TABLE IV-3. This is done by taking each condition where a grid signal, as for example a_1 , is true (1), and then expressing in terms of logical clock products of A_1 and A_2 the conditions that made this possible. Thus, the complete logical equation for each grid proposition is the logical sum of all such true products, and the clock proposition for timing purposes.

$$\begin{aligned}
 a_1 &= (A_1' A_2' + A_1' A_2) C \\
 o a_1 &= (A_1 A_2 + A_1 A_2') C \\
 a_2 &= (A_1 A_2' + A_1' A_2) C \\
 o a_2 &= (A_1 A_2 + A_1' A_2') C
 \end{aligned}$$

A network created according to these four logical equations would function as the desired binary counter, but it would be wasteful of logical elements. By

simplifying these equations, using expressions developed by Boolean algebra (see "Basic Symbolic Logic" in APPENDIX for discussion of Boolean Algebra), the following logical equation can be obtained:

$$\begin{aligned}
 a_1 &= A_1' (A_2 + A_2') C \text{ but} \\
 A_2 + A_2' &= 1.
 \end{aligned}$$

Hence, $a_1 = A_1' C$;

and $o a_1 = A_1 (A_2 + A_2') C = A_1 C$.

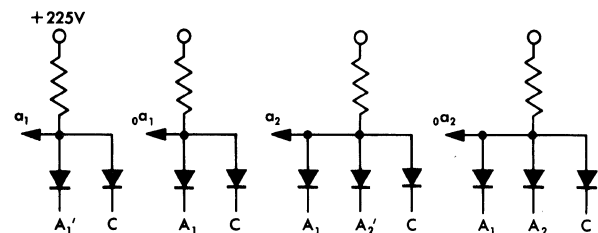
Then, $a_2 = A_1 A_2' C$;

and $o a_2 = A_1 A_2 C$.

The maintenance engineer should acquire some familiarity with such logical manipulations because it may occasionally be necessary for him to use logical equations and understand their function.

These equations may appear in different forms depending upon the requirements of the over-all design logic. It is important to be able to recognize the commonly used equations for arithmetic manipulations, for example, such are used in counting, adding, subtracting 1 from a number, etc.

Using the logical equations developed above, the schematics for the logical network may be drawn as follows:



Note that the flip-flops themselves are not shown, which is the usual practice on the logical schematics in this manual; but the connections would follow the pattern of the block diagram shown previously.

As a final step, the waveforms which would appear at various significant points in the circuit of the binary counter can be predicted, as shown in Figure IV-8.

A similar binary counter is actually used in the Model 102-A Computer to generate the three pulses (P_0 , P_1 , and P_2) required for timing the three bit positions in each octal digit of a word. The circuitry for this pulse counter is shown in Figure 7, Vol. II. Here the combination of unclamped flip-flops (located at TC4-A11 and TC4-B11) and flip-flop drivers is used in place of the logical flip-flops A_1 and A_2 assumed in the example of the binary counter considered previously. The computer's pulse counter consists of clock products with certain logical propositions used to operate two flip-flops, B_1 and B_2 . The output matrix of the two flip-flops develops the three pulses. Here P_0 is the logical product of B_1' , $B_2' = P_1$, and $B_2 = P_2$.

One point worth noting is that the grid propositions include terms which are effective only when the pulse counter is first started. Thus the product ZM_s , for example, is the logical product of the "CLEAR" proposition Z (obtained when the "CLEAR" button on the computer control panel is pressed) and the synchronizing pulse M_s . After having been used once, these terms are no longer required for the continued operation of the pulse counter. Thus the logical product for grid propositions ${}_ob_1$ and b_2 contains the negation of ZM_s since $(ZM_s)' = Z' + M_s'$; and ZM_s appears merely as part of the logical sum for grid propositions b_1 and ${}_ob_2$ and goes false immediately after its use during starting.

Remembering that this pulse counter counts only three pulses and then recycles, unlike the counter previously

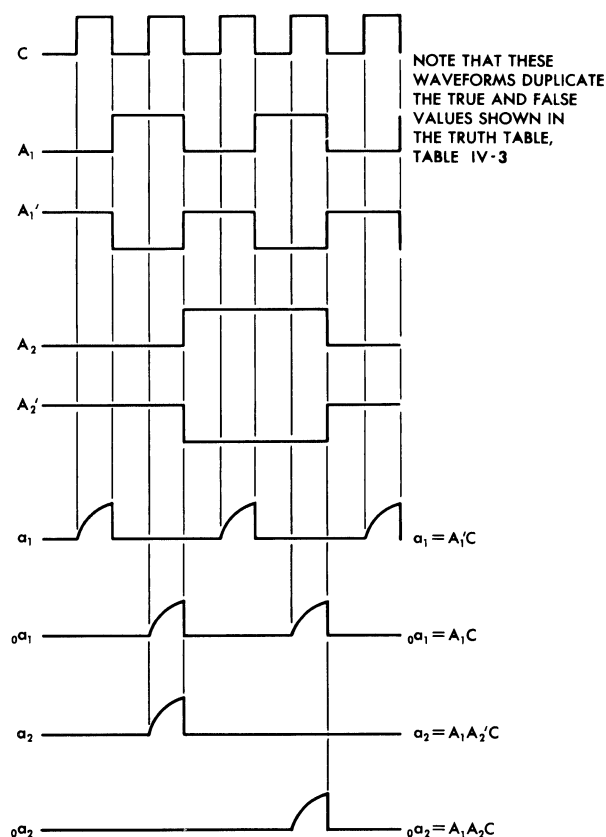


Figure IV-8. Predicted Waveforms

considered, the truth table would be written as indicated in TABLE IV-4.

Time	B_2	B_1	b_1	${}_ob_1$	b_2	${}_ob_2$
P_0	0	0	1	0	0	0
P_1	0	1	0	1	1	0
P_2	1	0	0	0	0	1

TABLE IV-4. Truth Table for Pulse Counter

From this truth table, as before, the following logical equations are derived:

$$\begin{aligned}
 b_1 &= B_1'B_2'C = P_0C \\
 {}_ob_1 &= B_1C = b_2 \\
 {}_ob_2 &= B_2C
 \end{aligned}$$

Note that these logical equations correspond exactly to the circuit design in Figure 7, Vol. II, except that Z and M_s are ignored since they are present only at the start of computation. In these equations, P_0 is defined as $B_1'B_2'$; $P_1 = B_1$; and $P_2 = B_2$.

The waveforms at significant points will appear as shown in Figure IV-9.

A truth table constructed for the octal digit counter is shown in TABLE IV-5. In the schematic for this counter (see Figure 7, Vol. II), the term $P_2^0_{13}$ appearing in the logical products required for grid propositions o_{d_3} and o_{d_4} is obtained from the octal digit matrix and equals $D_1D_2'D_3D_4B_2$, where $P_2 = B_2$.

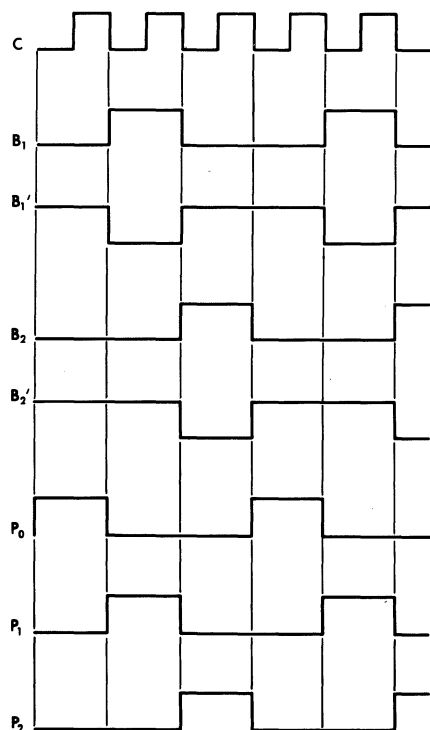


Figure IV-9. Idealized Waveforms of Pulse Counter

Note that although the four binary stages could count 16 steps, this digit counter is inhibited to recycle after the 14th

digit.

From TABLE IV-5, the equation for any octal digit can be obtained immediately by forming the appropriate products of the D propositions and their primes corresponding to each entry in the column headed "Octal Digit".

The logical network developed from the logical equations which can be derived from TABLE IV-5, using similar logic to that previously used for the binary counter and the pulse counter, consists of an array of logical products. For example, a typical product is $D_1'D_2'D_3'D_4'$. The array of diodes and resistors used to generate a complete set of such products is called a matrix. The digit counter matrix is driven by the D flip-flops which generate the 14 separate octal digit propositions. Each of these digit propositions goes true at a particular time corresponding to the position of that digit in the sequence of a word (see the waveforms in Figure IV-10). The grid propositions, unclamped flip-flops and flip-flop drivers, and matrices appear in Figure 7, Vol. II.

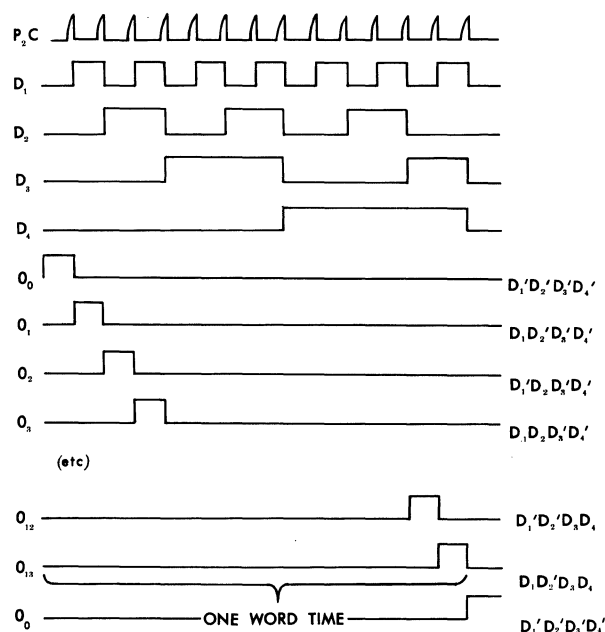


Figure IV-10. Idealized Waveforms of Digit Counter

Existing States				Grid Propositions to achieve Next Desired State								Octal Digit Time
D ₁	D ₂	D ₃	D ₄	d ₁	_o d ₁	d ₂	_o d ₂	d ₃	_o d ₃	d ₄	_o d ₄	
0	0	0	0	1	0	0	0	0	0	0	0	0 ₀
1	0	0	0	0	1	1	0	0	0	0	0	0 ₁
0	1	0	0	1	0	0	0	0	0	0	0	0 ₂
1	1	0	0	0	1	0	1	1	0	0	0	0 ₃
0	0	1	0	1	0	0	0	0	0	0	0	0 ₄
1	0	1	0	0	1	1	0	0	0	0	0	0 ₅
0	1	1	0	1	0	0	0	0	0	0	0	0 ₆
1	1	1	0	0	1	0	1	0	1	1	0	0 ₇
0	0	0	1	1	0	0	0	0	0	0	0	0 ₈
1	0	0	1	0	1	1	0	0	0	0	0	0 ₉
0	1	0	1	1	0	0	0	0	0	0	0	0 ₁₀
1	1	0	1	0	1	0	1	1	0	0	0	0 ₁₁
0	0	1	1	1	0	0	0	0	0	0	0	0 ₁₂
1	0	1	1	0	1	0	0	0	1	0	1	0 ₁₃
0	0	0	0	1	0	0	0	0	0	0	0	0 ₀ (next cycle)

TABLE IV-5. Truth Table for Octal Digit Counter

In presenting the waveforms for the octal digit counter, the grid propositions, d_1 , o_1 , etc., have been omitted, since they can be readily developed as in the previous examples. Also the primes of the digit propositions have been omitted, since their waveforms are apparent as the negations of the propositions themselves. The clock product (P_2C) is shown to indicate its timing function, signaling the end of each octal digit; to demonstrate its comparative length in time, since there are three pulses corresponding to binary digits in each octal digit; and to re-

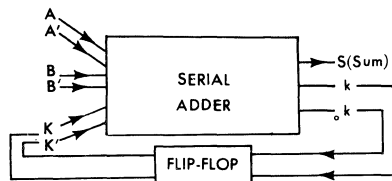
emphasize that the pulse counter is driven by the clock, and that the pulse counter drives the digit counter.

Note that the octal digit matrix makes it possible to pick off any individual octal digit time, or group of octal digit times, and use this to control the time of other logical operations. For example, as can be seen in Block 10 (see Block Explanations, Vol. II), the output of the H register (H_2) is connected to the input of the G register (G_0) during 0_{0-3} *. The period of these four digit times is controlled by the digit

* This notation is an abbreviation for the proposition, $0_0 + 0_1 + 0_2 + 0_3$, the logic sum of these four digit times.

matrix. Similarly the pulse and octal digit matrix (see Figure 7, Vol. II) makes it possible to select any pulse time of any digit time. For example, in the same block it is desired to apply a certain proposition to k at P_{10}^{013} , the timing control comes from this combined matrix.

The Serial Adder: The serial adder provides another useful example of how logical circuitry is developed to perform a certain operation for which a truth table can be constructed. As shown in the block diagram below, the serial adder has three pairs of inputs, A and A' (addend), B and B' (augends), K and K' (carry from previous pulse time). The outputs are S (sum) and the grid propositions (k and ${}_o k$) for the carry for the next digit.



The rules for this adder network are as follows. If there are no 1's in the unprimed inputs, the sum output should be 0 and the carry false grid (${}_o k$) should be set true. If there is a single 1 among the unprimed inputs, the sum output should be 1 and the false grid (${}_o k$) should be set true. If there are two 1's among the unprimed inputs, the sum output should be 0 and the true grid (k) should be set true. If the unprimed inputs are all 1's, the sum output should be 1 and the true grid (k) should be set true. The truth table for this serial adder is shown in TABLE IV-6. In considering this table, keep in mind that S is the arithmetic sum of A , B , and K , not the logical sum; and that in binary addition, the sum of two 1's gives a 0 and a carry 1. In an addition, the states of A , B and K may occur in any sequence, depending on the numbers be-

ing added. They do not necessarily progress in the order shown in the truth table. Compare this with the octal digit counter where exact sequence was essential.

POSSIBLE INPUTS			DESIRED OUTPUTS		
A	B	K	S	k	${}_o k$
0	0	0	0	0	1
0	0	1	1	0	1
0	1	0	1	0	1
1	0	0	1	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	1	0

TABLE IV-6. Truth Table for Serial Adder

Note that in TABLE IV-6 as was previously explained in connection with TABLE IV-3, the condition of the K flip-flop will stay in its stable state unless a change is required by the conditions of the addition.

Assuming the following two binary numbers represented serially by the time-varying logical propositions (A and B) and based on the truth table, the waveforms and the condition of the K flip-flop would appear as shown below.

A 0 0 1 1 0 1 1 0 0 1 0

B 1 0 1 1 1 0 0 0 1 1 1

K 0 1 1 0 0 0 0 1 1 0 0

S 1 1 1 0 1 1 1 1 0 0 1

From TABLE IV-6 using the same techniques previously employed in constructing the logical equations, the following can be written for the serial adder:

$$k = ABK'C \quad \text{and} \quad {}_o k = A'B'KC$$

$$S = A'B'K + A'BK' + AB'K + ABK$$

These equations, in turn can now be expressed in terms of the design of a logical circuit, which will take the form shown in Figure IV-11.

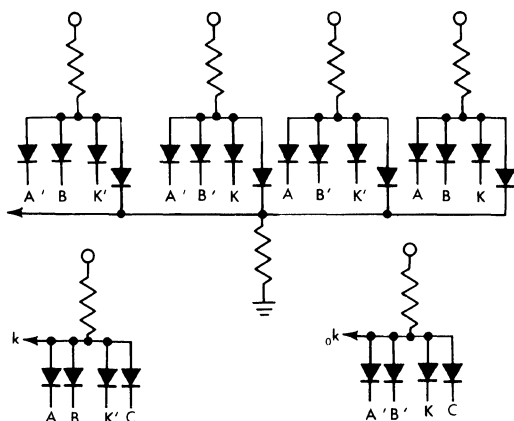


Figure IV-11. Logical Circuit for Serial Adder

Another example of logical design is the extract function in the Model 102-A computer. The extract command moves selected binary digits from one word into the same positions of another word according to the presence of 0's (zeros) or 1's in an intermediate word. The command is set up in the form

$$\text{ex } m_1 \ m_2 \ m_3$$

In the operation, those bits in (m_1) which are in the same positions as the 1's of (m_2) are put into the corresponding positions of (m_3). Where the bits of (m_2) are 0's, the corresponding positions of (m_3) remain unchanged. The contents of m_1 are placed in the E register, (m_2) in F, and (m_3) in G. As all results in the computer must appear in E for putaway, the machine's extract operation is performed as follows. A bit is selected from either E or G, according to whether the corresponding bit in F is a 1 or a 0, and then the result is put in E. The following is a numerical example of this operation:

(m_1) E 35010111001101010011 etc.
(the entire 42 bits are not shown)

(m_2) F 00000111100010100111 etc.

(m_3) G 25001000111010011000 etc.

E 25001111011000011011 etc.

(at end of extract operation)

This operation is expressed in TABLE IV-7.

E_5	F_2	G_5	E_0	
0	0	0	0	Where E_5 is the output of the E-line and represents the bits that appear in (m_1)
0	0	1	1	
0	1	0	0	
0	1	1	0	F_2 is the output of F the bits of (m_2)
1	0	0	0	
1	0	1	1	G_5 is the output of G the bits of (m_3)
1	1	0	1	
1	1	1	1	E_0 is the input to E after the extract operation has been performed.

TABLE IV-7. Truth Table for Extract Circuit.

From this truth table, the following equation can be derived for use in designing the logical circuitry to carry out this extract operation.

$$(1) \ E_0 = E_5'F_2'G_5 + E_5F_2'G_5 + E_5F_2G_5' + E_5F_2G_5$$

The logical circuitry for this equation is shown in Figure IV-12.

Apparently this circuit would require 16 diodes. However, the logical equations can be simplified as was done previously for the binary counter.

Factoring

$$(2) \ E_0 = F_2'G_5(E_5' + E_5) + E_5F_2(G_5' + G_5)$$

But it is always true that

$$E_5' + E_5 = 1 = G_5' + G_5$$

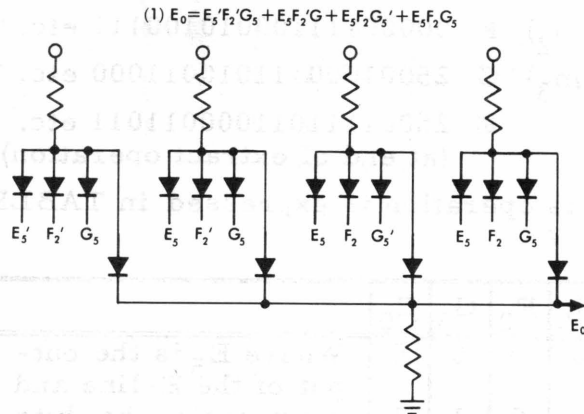


Figure IV-12. Logical Circuitry for Extract Operation

Hence the simplified logical equation reduces to

$$(3) E_0 = F_2'G_5 + E_5F_2$$

This equation is expressed by the logical circuitry shown in Figure IV-13.

Now the circuit design has been simplified

from its original form, requiring 16 diodes, to the form above requiring only six diodes. Such economical logical design is typical of the Model 102-A Computer. It also illustrates why it is important for the maintenance engineer to be able to recognize logical manipulation.

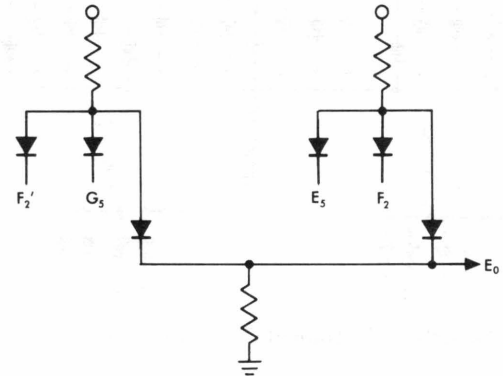


Figure IV-13. Logical Circuitry for Extract Operation (simplified)

MEMORY ELECTRONICS

DRUM AND HEADS

The memory drum, briefly described previously in SECTION III, is the storage device used in the 102-A Computer. The surface of this drum is coated with magnetic iron oxide. The drum revolves on a vertical axis inside a cylindrical housing called the "barrel". Mounted in ports suitably spaced around the circumference of the barrel are 35 electromagnetic heads for recording information on the drum, playing back previously recorded information, or erasing recorded data. Figure IV-14 shows the drum and barrel assembly removed from the computer.

The position of these 35 heads is shown in Figure 24, Vol. II. Note that there are 16 read-write heads for the main memory channels, one for each channel. Each of these heads has two windings, one for recording (writing) and

the other for playback (reading). Functioning of these magnetic heads is similar to the operation of those used in conventional tape recording. A pulse

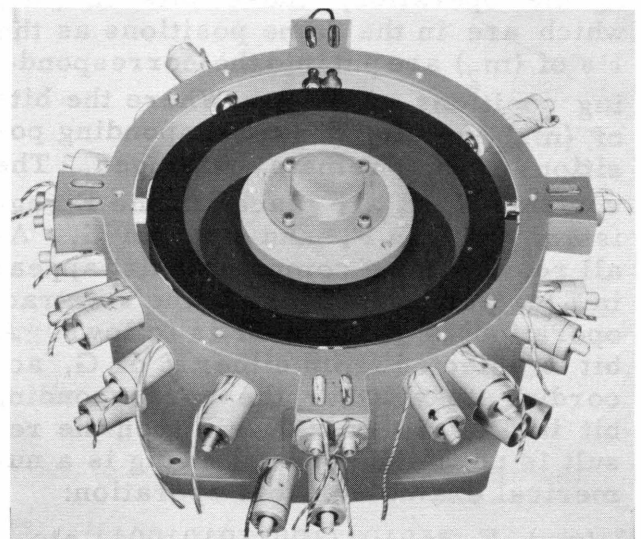


Figure IV-14. Memory Drum Assembly

of current in one direction through the record winding of the head, for example, results in magnetizing a small area of the drum surface in a corresponding direction.

Figure 15, Vol. II, also shows 15 heads for the five recirculating registers, E, F, G, H, and J. Each of these registers utilizes three separate heads, each with a single winding, for reading, writing, and erasing. For instance, E_{PB} is the playback head of the E line.

The remaining four heads include two for the clock channels, C_1 and C_2 , a "working" clock used to supply timing pulses for the computer, and a duplicate. These two clock channels are recorded initially at the factory, and the duplicate is provided merely as a safeguard in case the working clock is accidentally damaged. These special clock heads normally are used only for reading.

The word channel (M_W), used for identifying the 64 sectors^W of the memory, has a read-write head with two windings. However, normally this channel is numbered sequentially and then the write-winding of the head is disconnected, so that the head is used only for reading sector positions. If at any time it is desirable to renumber the sectors on the word channel for minimum access time, the write-winding again is connected for the renumbering and later disconnected.

FUNCTIONAL DESCRIPTION OF MEMORY ELEMENTS

Main Memory

As mentioned previously, the 16 channels which comprise the main memory are so placed as to come under the 16 read-write heads shown in Figure 24, Vol. II. Each channel provides storage space for 64 words of 42 bits each, or 2688 bits. Since the diameter of the memory drum (and hence of a channel)

is 12 inches and its circumference 37.68 inches, there are about 71 bits per inch. The width of a channel is about 0.2 inch.

Each word-space on a channel is called a cell or sector, and each of the 1024 cells in the main memory has an address, specified by four octal digits. The first two of these digits identify the channel, with the channels numbered octally from 00 to 17. The last two digits specify the sector (cell), with each sector one word in length, hence 64 sectors per channel. Sectors are numbered octally from 00 to 77. Thus, main memory addresses run from 0000 to 1777.

Since the memory drum revolves at about 40 rps, one revolution requires about 25 milliseconds. Hence, if a main memory cell has just passed under a read-write head, it will be available for reading or writing 25 ms later. If the cell is about to pass under the head, access time will be a fraction of a millisecond. Thus, the average access time to any cell will be half of the maximum time of 25 milliseconds, or 12.5 ms. The time to read or write one word is about 0.4 ms.

Recirculating Registers

There are four one-word recirculating registers located on a single channel on the drum (see Figure IV-15). These registers, the E, F, G, and H lines, act

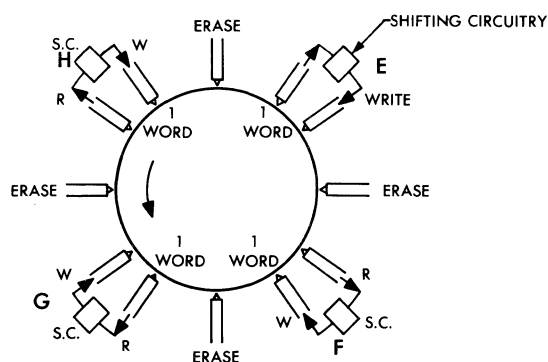


Figure IV-15. One-Word Recirculating Registers

as one-word storage units for the arithmetic and control elements of the computer. In general terms, the functions of these registers are as follows:

- E - Arithmetic unit register; and intermediate storage for information during input and output.
- F - Arithmetic unit register.
- H - Control register.
- G - Dual-purpose register, used both as a control register and as an arithmetic unit register.

These functions are delineated in more detail in the sample program routines under "Examples of Computer Operation" in this Section.

In Figure IV-16, the diagram of the E register and its associated separate read and write heads will serve to illustrate its operation. Information held in this register is recirculating constantly while the computer is operating. Remembering that the E line is defined as a one-word register, it can be understood that as the drum rotates, the bits of this word are read off, one by one, at the read head and recorded back on the drum through the write head. The word in a register will continue to recirculate in this manner until it is written over by new information entering the register, or unless the line is temporarily broken so that the erase head can clear the line, filling it with zeros.

Now, a closer examination of the recirculation process, as diagrammed in Figure IV-16, reveals that during this process 36 of the 42 bits of the word are on the drum surface at any given instant, while six bits are held in flip-flops outside the drum in the line between the read head and the write head.

Observe that at the end of one word time, or 42 pulse times, the word will have gone around this closed loop one time. Bit P_0^0 will have gone from E_5 through the E write circuit, along the

surface of the drum to the E read head, and through the E flip-flops back to E_5 .

These flip-flops associated with the recirculating registers are called "stepping flip-flops", and they are driven by clock pulses, synchronized with the rotation of the drum so that each bit progresses along the flip-flop chain at the same rate as on the surface of the drum. There are five stepping flip-flops, plus a write flip-flop, in the E and G lines (as shown in Figure IV-16); and two stepping flip-flops plus a write flip-flop associated with each of the F, H, and J (buffer) registers.

To introduce new information into a register, the chain of flip-flop stages is broken and new data is fed into the line input. For instance, the E_5 - E_0 con-

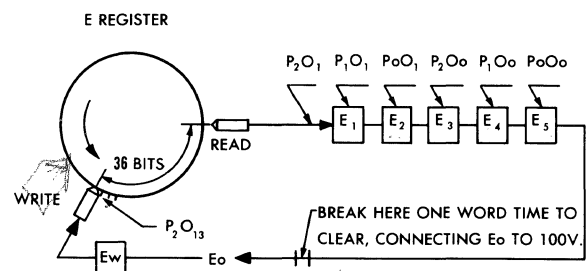


Figure IV-16. Diagram of E Register

nection is broken, additional flip-flops (one or more of the A_1 through A_4 flip-flops) are inserted between E_5 and E_0 for one or more word times, and the bits from these A flip-flops are fed into E_0 . Note that E_0 is a proposition, not a flip-flop (see Figure IV-17).

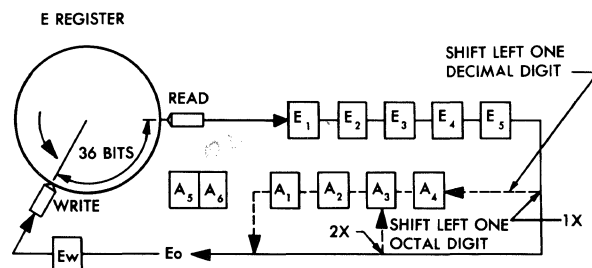


Figure IV-17. Inserting Information into the E Register

At this point, a brief description of the arithmetic flip-flops, A_1 through A_{12} is pertinent. To shift information into one of the recirculating registers, the E line, for instance, its chain of flip-flop stepping stages is expanded by inserting one or more of the flip-flops A_1 through A_4 . The state of the A_6 flip-flop, which is preset by the "o" or "d" key on the Flexowriter, tells whether the fill is octal or decimal. A "0" in A_6 means that an octal number is being filled; and A_1 through A_3 will be utilized to carry the three bits for each octal digit, as shown in Figure IV-17. A "1" in A_6 indicates decimal fill, and the A_1 through A_4 flip-flops are inserted in the E line to bring it the four bits for each decimal digit. Details of the use of the A flip-flops are discussed under "Flexowriter Circuitry and Operation" in this Section; and also may be observed from studying the appropriate Block Explanation Forms in Vol. II.

It is important to note that lengthening the E line by n flip-flops for one word time shifts the previous contents of E left by n bits and fills in the n bits stored in these added A flip-flops at the right end of the E word. Conversely, to shift the contents of E to the right, the chain of delay flip-flops (E_1 through E_5) is shortened temporarily by passing up to four of these flip-flops. Thus E_1 can be connected to E_0 , for instance, and this shifts the contents of E to the right by four bits. This is shown graphically in Figure IV-17.

Another point worth noting is that any of the recirculating registers can be connected to any of the other registers for part or all of a word time. For example, note in Block 17 (see Block Explanations in Vol. II) that the G line is connected to the H line (G_5 to H_0) for the period 0_4 through 0_7 so that the m_2 portion of the word in G is read into H.

The Buffer Register: This register deserves special attention because it is an eight-word recirculating register located physically on its own channel of the memory drum. It has two principal functions:

(1) It serves as a convenient transfer device between the computer main memory and such input-output units as the Model 126 Magnetic Tape Handling Unit and the IBM Summary Punches. That is, information from the main memory is transferred to the J (Buffer) register and then read from that register into a magnetic tape unit for storage.

(2) It provides an extension (volatile) to the main memory for quick access. This is because all eight words of the buffer register pass the J read and write heads exactly eight times per drum revolution. Thus maximum access time to the buffer is about 3 ms. and average access time is only 1.5 ms.

Figure IV-18 diagrams the read-write cycle of the buffer register. Note that there are 333 bits on the drum surface at any one time, and three bits contained in the J flip-flops (one each in J_1 and J_2 , and the third in J_w). The total is

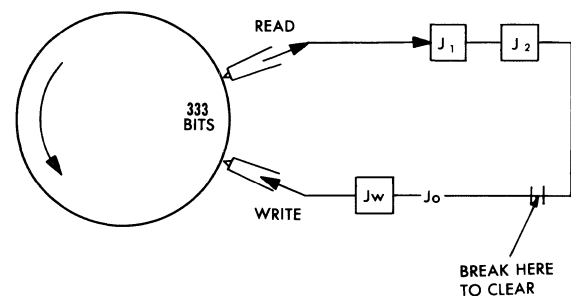


Figure IV-18. Buffer Register

336 bits (8 words x 42 bits per word). The buffer cells are numbered from 2000 through 2007, and the last digit is synchronized with the corresponding digit in the word channel, M_w . These digits 0 through 7 of the buffer register

address are, accordingly, correlated with the least significant digit of the sector addresses in any given main memory channel.

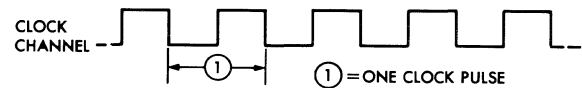
The two special addresses, cells 2100 and 3000, are useful also in programming. Cell 2100 has no physical reality as a cell on the memory drum. Rather, it is logical circuitry so designed that it can be used only in the reading or playback operation, when zeros are to be read into a particular register. What happens, when Cell 2100 is called for, is that the input to the E line (E_0), for instance, is made false for one word time so that the computer now reads zeros into the E line for one word. Thus cell 2100 is occasionally called a "zero access-time zero" device. It is important to note that since Cell 2100 does not exist as a cell, there can be no put-away in it; it is a circuit for effectively filling a register with zeros, which then can be utilized in any way desired by the programmer.

Cell 3000 is a special designation for the G register, again used only for look-up, not put-away. Controlling conditions are that if Cell 3000 is called for in m_1 of H, the contents of G are transferred into E; if it is called for in m_2 of H, (G) goes into F; if the command calls for Cell 3000 in m_3 of H, put-away is actually in the buffer cell 2000.

Clock Channels

There are two clock channels, C_1 and C_2 , as previously mentioned, used for regulating the timing of all logical operations in the computer. Each of these channels contain 64 words of 42 pulses each, or 2688 clock pulses. A clock pulse, which corresponds to one bit-space on the other channels, contains

both a 0 and a 1 representation as follows:



This clock waveform is a square wave with a frequency of $64 \times 42 \times$ (approx.) 40 (rps) = 107, 520 cps. (approx.). A master oscillator operating at this frequency is used to record the clock on the C_2 (duplicate) channel. From this C_2 recording, the C_1 channel is now recorded. The word channel, as described in the following paragraph, is written (recorded) using C_1 for timing. Thus if at any time the C_1 "working" clock is damaged, it can be recorded again from C_2 , as before, without any necessity for changing the timing of the word channel. This entire procedure for initially recording the clock channels is, of course, a factory operation. Normally, after the recording, the windings of the C_2 head are disconnected. If it becomes necessary to record C_1 from C_2 in the field, the complete procedure is provided in SECTION VI.

Word Channel

This channel, designated as M_w , is used for locating the sectors of the main memory channels and the buffer register. M_w contains 64 words of the form

00	0000	0000	0000	to
00	0077	0077	0077	

Figure IV-19 indicates the appearance of the word channel, showing only the last two octal digits (for the sake of brevity) in each word space.

The first two digits of an address are

used to select the correct channel, and then the last two address digits are compared automatically with the output of the word channel read circuits. When the word channel digits coincide with the sector address digits, the next word appearing in the selected channel is made available for reading or writing.

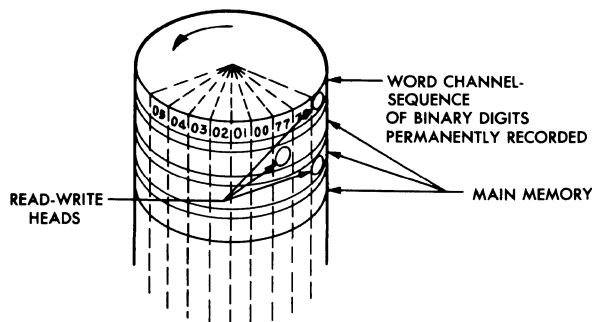


Figure IV-19. Word Channel on the Drum

It is the next word because it is necessary to allow for the time required to read or write a word, as indicated on Figure IV-20.

Numbering of the word channel may be in any order around the drum, so long as each consecutive group of eight M_w

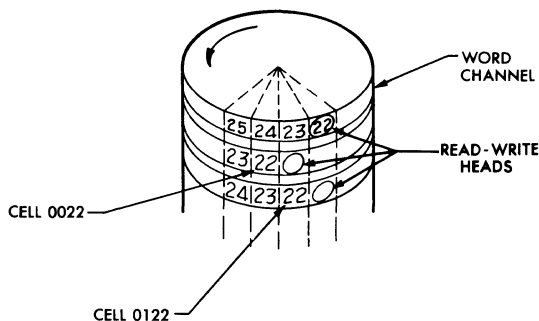


Figure IV-20. Locating An Address on the Drum

words contains the digits 0 through 7 in the same order in the LSOD position of each of the three sector addresses. This must be done to permit proper access to the eight-word buffer when desired.

The word channel is recorded at the

factory but can be changed. A special paper tape is prepared for filling the word channel from the Flexowriter; the word channel write head and other parts of the M_w record circuit are properly connected, and the "Synchronizing Pulse and Word" switch (TS3) is turned on. After the word channel is recorded, its performance in sector selection is carefully checked with both typical main memory channels and the J line (Buffer Register).

Synchronizing Pulse

This is a single pulse recorded in the P_{12}^0 position of all sectors of the word channel. This single pulse is used for synchronizing the pulse and octal digit counters when the computer is first turned on. To test its operation, the sync pulse is checked for coincidence with P_{00}^0 when the clear button is depressed (proposition ZM_s at $4(52V)P$).

MAGNETIC TECHNIQUES

As previously mentioned, the surface of the memory drum is coated with magnetic iron oxide. In recording pulses on this drum, the direction of the applied magnetic field is tangential to the circumference of the drum; and the field is obtained from the record winding of an electromagnetic head.

Each magnetic head is assembled from two ferrite-core pole pieces, as shown in Figure IV-21. A .001-inch silver shim is used to provide a gap at one end. Recording and/or playback coils are wound on the legs of the pole pieces. The end of the head containing the shim is slightly less than 0.25 inch wide. Each head is placed so that the gap is parallel to the vertical axis of the drum. The space between the face of the head and the drum surface is adjusted to a distance of about .001 inch.

When the drum is rotating and sufficient current is passed through the recording

head, a strip of the drum surface will be magnetized to saturation in a circumferential direction. Reversal of the direction of current flow through the coil will reverse the sense of the surface magnetization as shown in Figure IV-21.

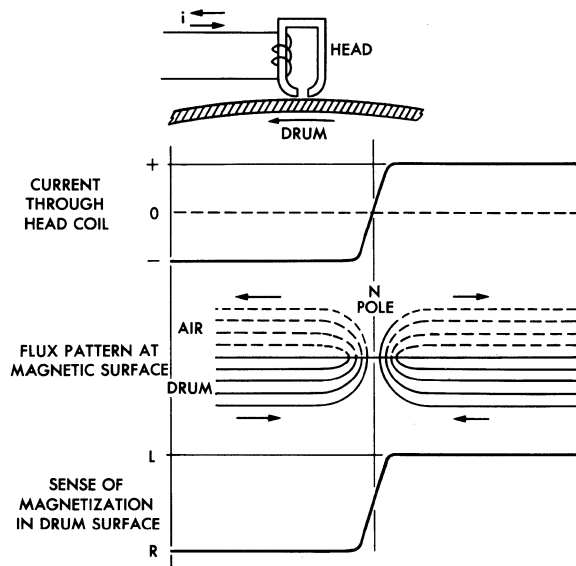


Figure IV-21. Magnetic Head

A series of reversals of the current through the recording coil on the head might produce the typical patterns shown in Figure IV-22.

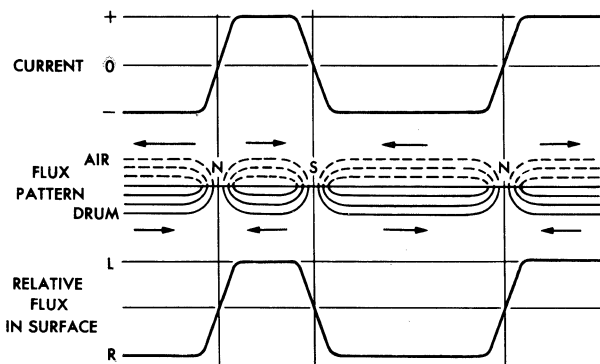


Figure IV-22. Typical Record Pattern

This type of magnetic recording is called "non-return to zero" because the

drum surface is magnetized to saturation in one direction or the other over the entire length of the channel under the recording head.

Now if the drum surface, which has been magnetized in the pattern just described, is assumed to move past another magnetic head which has a playback (read) winding, a voltage will be induced in this coil that is proportional to the rate of change of flux at the gap. As indicated, the flux is relatively constant except at the poles (N and S) on the drum surface. When one of these poles passes the head, a voltage pulse will be induced in the read coil; the polarity of this pulse can be assumed to be negative for a north pole and positive for a south pole, as shown in Figure IV-23.

The amplitude of such a voltage pulse observed in the read heads of the Model 102-A Computer is about 0.6v peak.

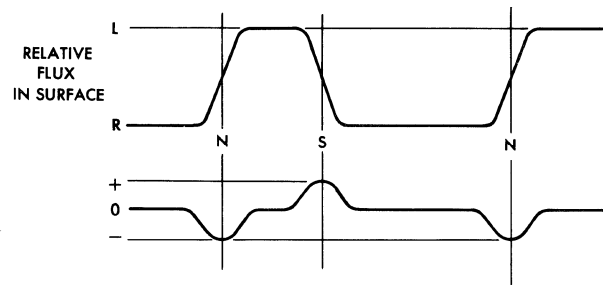


Figure IV-23. Pulse Patterns Induced in a Read Head

If one sense of the flux is associated arbitrarily with the recording of a binary 1, and the other sense with 0, then these voltage pulses obtained at the read head may be converted into a form suitable for triggering a bi-stable element which will then reproduce the information recorded in the memory at the logical voltage levels.

ELECTRONIC TECHNIQUES

Recirculating Registers

The electronic circuitry of the recirculating registers is shown in Figures

15, 16, 17, 18, and 19 in Vol. II. These lines all have separate heads for recording, playback, and erasing. The record head coil is in the plate circuit of one triode of a 5687 record tube (20-267004) as shown in Figure 21 (2 of 3) in Vol. II. This double-triode tube will serve as the record tube for two one-word recirculating lines.

In recording from the E line, for example, the record tube (half of 5687) is turned on and off by proposition E_w , which is the output of the clamped flip-flop whose inputs are $E^o C$ and $E^o C$. The simplified schematic in Figure IV-24 illustrates this recording operation.

As the recording must be of the non-return-to-zero form, and only 1's are recorded by the record tubes of the recirculating lines, an erase head is located a few sectors away from each playback head. A steady current is maintained in the erase heads in the "zero record" direction so that the information content of any recirculating line will be all zeros until some ones are recorded through the record tube. Breaking the recirculation loop (see Figure IV-16) for one word-time (eight word-times for the buffer register, j) will clear the register, since no ones are recorded and all previous ones are replaced by zeros as they pass the erase head.

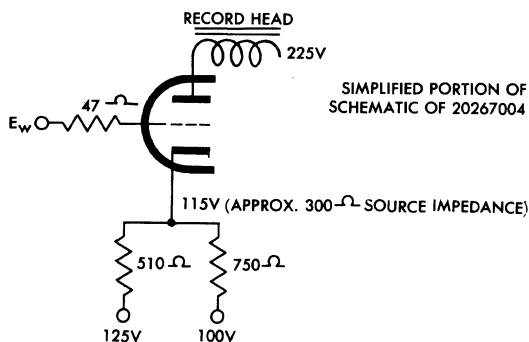


Figure IV-24. Simplified Schematic of 5687 Record Tube

The five erase heads are connected in series from +100 volts through a 2500

ohm, 5 watt, resistor to ground. The resistor is in the ground end of the circuit at MTC 1/2 B-9. The current through the head windings is approximately 38 ma, calculated as follows:

$$I = \frac{100v}{2500 + 5.28} = \frac{100v}{2640(\text{ohms})} = 38 \text{ ma,}$$

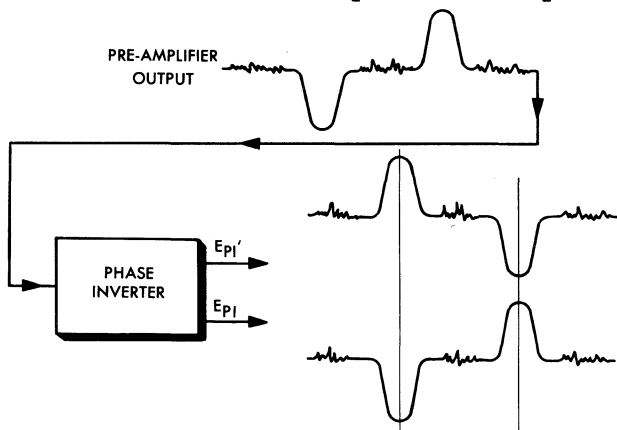
Where each of the five erase heads has a resistance of approximately 28 ohms. The erase heads plug into the appropriate recirculating line preamplifiers.

When a magnetic pattern corresponding to any given record proposition is moving past the playback head, the output of that head is used to generate a register output proposition of the same form as the original record proposition. The circuit for generating the output proposition includes a preamplifier, phase inverter, dual clipper, and a flip-flop. A special signal, the sampling clock, derived from the clock channel on the drum, is also required.

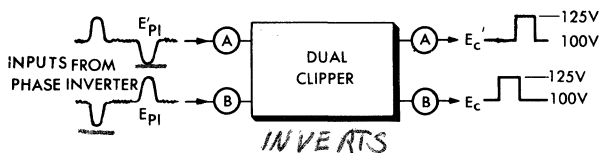
The output signal from the playback head is amplified in a two-stage feedback preamplifier (20267006) with a gain of about 100, shown in Figure 21 (2 of 3), Vol. II. Gain is controlled by potentiometer R7 in the second half of 6BQ7A amplifier stage grid circuit. The resistor across the playback winding is for damping. Output, from the plate of the second stage (terminal 8, is about 70 volts peak and is in phase with the head output because of the two-stage amplification. Typical waveforms are shown in Figure IV-25.

The output of the pre-amplifier drives a unity-gain split-load phase inverter (10667016) shown in Figure 21 (1 of 3) in Vol. II. From the plate of one-half of the 5687 double triode an inverted signal is obtained--output A', for example, from input A--while the cathode supplies an output (A) in phase with the input. Thus, the two outputs from the phase inverter are equal in amplitude but opposite in phase, as shown in the

waveforms. The output of this phase



inverter is fed into a dual clipper (106-67017); the circuit is shown in Figure 21 (1 of 3), Vol. II. Assuming the output E_{PI} of the phase inverter to be fed into input A of the dual clipper, and output E_{PI}' to go into input B.



Note that the grids of the 6BQ7A in the clipper normally have a positive bias, with both sides of the tube conducting. Thus, a positive input signal will have no effect on the clipper output. However, a negative signal of sufficient amplitude applied at input A, say, will make the cathode of diode D8 go negative and this in turn will reduce the plate potential of D8 so that the grid (pin 7) goes negative and cuts off the tube. The clipper's input circuit prevents dc shift and balances impedance. The consequent rise in plate voltage (pin 6) is clamped by diodes in the output, and thus is limited to a rise from +100v to +125v, the logical voltage levels. Since the input negative pulse is about -70v in amplitude, the action of the clipper inverts this signal and clips off the top, tending to square the pulse as shown in the waveforms.

These two outputs from the clipper, E_C and E_C' , next form logical products

with the sampling clock signal, C_S .

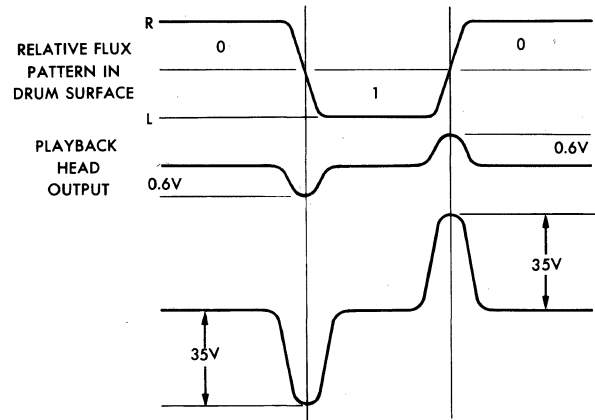


Figure IV-25. Typical Waveforms in the Playback Circuit

The sampling clock consists of a series of positive pulses, each of about 1.3 microseconds duration, whose trailing edges are synchronized with the trailing edges of the computer clock, C . The logical products of the clipper outputs and sampling clock serve as the grid propositions of the first recirculating register output flip-flop, for example, E_1 as shown on the waveforms in Figure IV-26.

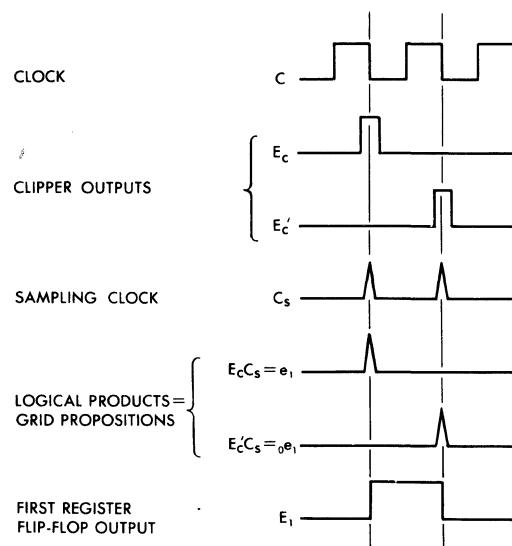


Figure IV-26. Clipper Output Waveforms

The composite chart of waveforms (see Figure IV-27) shows the principal steps

in the cycle of recording and playback of a recirculating register, and thus is a visual summary of the description on the preceding pages. Note that there is no phase lag between the output, E_1 , and the record (write) proposition, E_w .

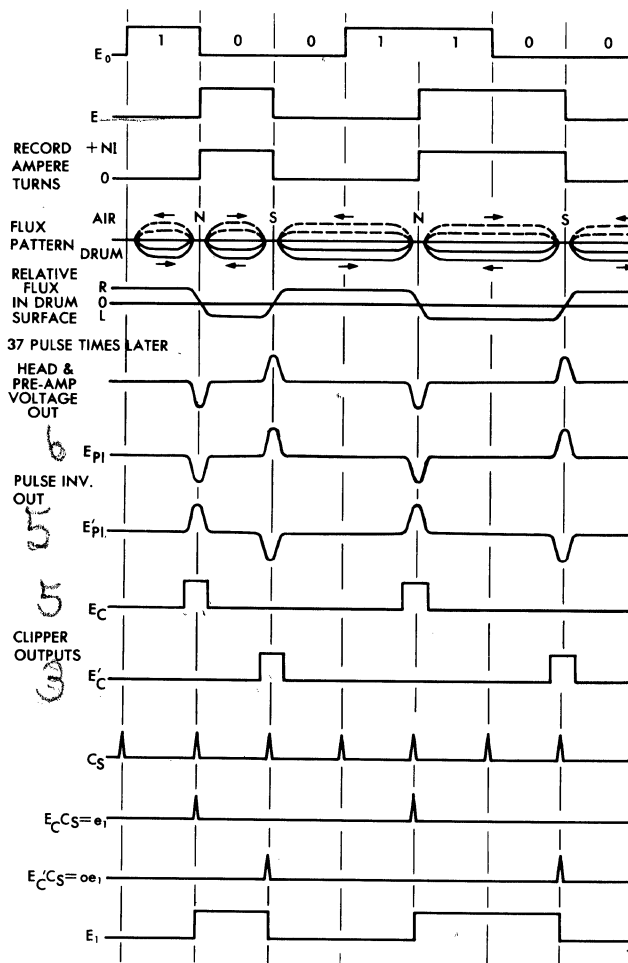


Figure IV-27. Idealized Waveforms in the Record/Playback Cycle of the E Recirculating Register

Now, still using the E register as an example, a typical train of pulses (a few bits of a word) may be followed through the five delay flip-flops in the E line. These waveforms appear in Figure IV-28. Note that in connection with clock products, there is a delay of one pulse-time caused by the logical product of the clock and the outputs of each flip-flop. This is shown graphic-

ally in the chart by the successive shifts to the right on the time scale of the bits 1 0 0 in traveling from E_1 through E_5 .

A similar set of waveforms could be drawn for the other recirculating registers, remembering that the E and G lines each contain five register (delay) flip-flops, while the F, H, and J lines each have two register flip-flops. Each recirculating register of course, also has one record flip-flop as described previously.

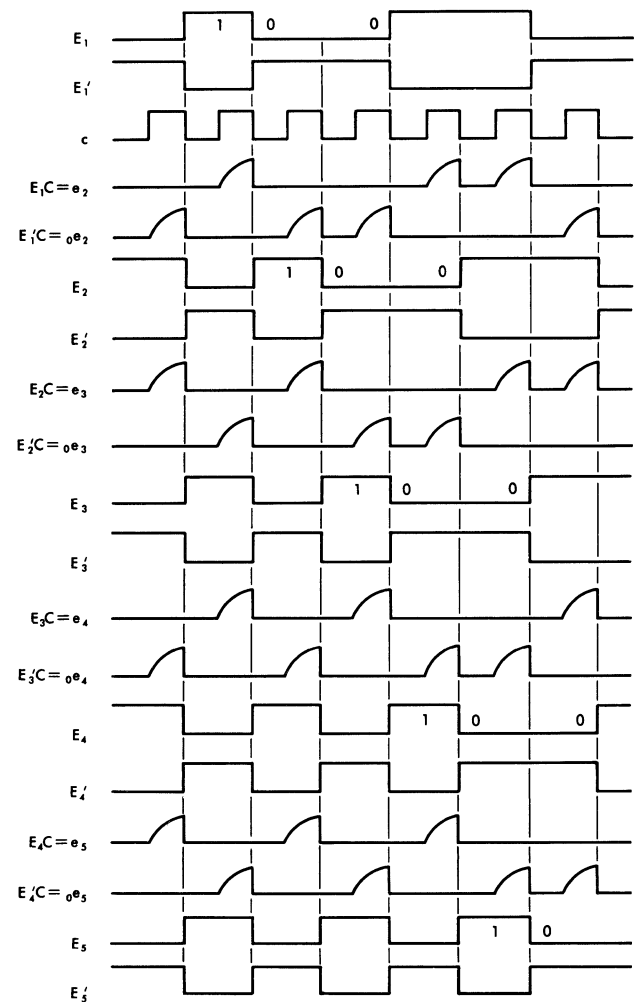


Figure IV-28. Idealized Waveforms in the E Flip-Flop Register

Main Memory Electronics
Recording into, and playback from, the

main memory channels are somewhat similar to the process described above for the recirculating registers, but there are numerous important differences in both procedure and circuitry.

Recording into a cell of any of the 16 main memory channels is accomplished by a transfer of information from either the E register or the J (buffer) register to that cell by means of the logical circuits shown in Figure 14, Vol. II, plus a record tube and the record (write) winding of the read-write head of the specific channel.

As shown on Figure 14, Vol. II, under the heading "Memory Record Propositions", there are two unclamped record flip-flops, R_1 and R_2 , and an associated flip-flop driver used to control the recording of information into the main memory. Flip-flop R_1 is set true for transferring the contents of the E register to the appropriate position in the memory; R_2 is set true for transferring (J) into eight continuous cells in the memory.

Note that grid proposition r_1 is the logical product of the following terms: C, the clock, for timing; K', the decision flip-flop (set false for the count to the next block); L_5 , one of the channel selector flip-flops (set false to indicate a putaway in the memory rather than in J); P_2O_{13} showing that recording starts at the beginning of the next word-time; $46 + 135$, a logical sum, indicating that either program counter 46 or PC 135 will make r_1 true. Similarly, r_2 is the logical product of five terms; and ϕr_1 and ϕr_2 are generated by logical sums and products. Details may be obtained by examining the Block Explanations for 46, 47, 135-137, 437 and 447, and the Flow Diagram in Vol. II.

At this point it might be well to note that the selection of channels in the main memory is accomplished by means of the flip-flops L_1 through L_5 , which are

set up to correspond to the first two digits (actually, the five least significant bits of these digits) of an address for look-up or putaway of information. TABLE IV-8 summarizes the information on channel selection.

Now, returning to a consideration of the memory record propositions in Figure 14, Vol. II, note that both a "True Record" and a "False Record" proposition are generated. The former means that a one is to be transferred from the E line or the buffer to the main memory, with the output of the E_5 or J_2 flip-flops true respectively. Similarly, "False Record" means that a zero from E_5 or J_2 is to be recorded in the memory.

Both of these record propositions form the inputs for the two special record tube drivers (20267365), each having two 6AN5 tubes in parallel, see Figure 21 (3 of 3), Vol. II. The input, of course, will be at one of the two logical voltage levels of +100 or +125v. The output is clamped, by the four clamping diodes shown, at the levels of +125 and +150v. Because of the negative grid bias, the two tubes in this driver are normally cut off and hence the output is +150v. The record proposition turns on the driver when the input signal level reaches +125v: That is, when the "True Record" proposition goes true, or when the "False Record" goes true the input signal level reaches 125v. The two 6AN5 tubes are used in parallel for this driver because of the relatively large current drawn here, about 50 ma per tube.

The true and false record propositions, inverted and placed at new voltage reference levels by the special record tube drivers, are then applied at the cathodes of the main memory record tube (20263329) shown in Figure 21 (3 of 3), Vol. II. Note that the plate of this tube is connected to the +415v supply instead of the +225v regulated supply that is used for most other plug-in units. The grids are connected in parallel to a

Bit Pattern	Logical Description	Octal Digits	Operation
L_5 L_4 L_3 L_2 L_1			
Writing into Memory (Putaway)			
B 0 X X X X	L_5 is 0	00 - 17 or 40 - 57	Select Main Memory Channel
B 1 B B B B	L_5 is 1	20 - 37 or 60 - 77	Select Buffer Register
Reading from Memory (Look-up)			
B 0 X X X X	L_5 is 0	00 - 17 or 40 - 57	Select Main Memory Channel
B 1 0 B B 0	L_5 is 1; L_4 is 0; L_1 is 0	20, 22, 24, 26, 60, 62, 64, 66	Select Buffer Register
B 1 0 B B 1	L_5 is 1; L_4 is 0; L_1 is 1	21, 23, 25, 27, 61, 63, 65, 67	Select "Cell" 2100
B 1 1 B B B	L_5 is 1; L_4 is 1	30 - 37 or 70 - 77	Select "Cell" 3000
Code: B - no significance; X - significant for specific channel			

TABLE IV-8. Bit Pattern for Channel Selection

channel selector proposition for each record tube. These channel selector propositions are obtained from the matrix shown in Figure 13, Vol. II, and are composed of logical products of the outputs of the channel selector flip-flops, L_1 through L_4 . The channels are numbered octally; thus, for example, channel 17 would be $L_4 L_3 L_2 L_1$ while channel 6 would be $L_4 L_3 L_2 L_1$, as indicated previously in Table IV-8.

When a channel selector proposition goes true, the grids of the memory record tube have the logical level of +125v

applied to them. However, the cathodes are tied to the +150v of the special record tube drivers, and remain at this level until one of these drivers is turned on by a record proposition. Then the driver outputs, R_T and R_F , range between +150 and +115v depending upon the content of the word being recorded. When a 1 is to be recorded, for instance R_T drops to +115v and hence the cathode of the left side of the 6BQ7 record tube drops to this level, this half of the tube conducts and current flow through the recording coil on the head in the direction B to A, resulting in the flux pattern

on the drum surface required for recording a 1. Now, when a 0 (zero) is to be recorded, R_F drops to +115v (R_T rising to +150v) and the same sequence of events takes place, with current flowing from B to C in the recording coil (in the opposite direction) so that the flux pattern for a 0 results.

In reading (playback) from a main memory channel, note that the grids of the record preamplifier, a 5963 double triode (20263329) shown in Figure 21 (3 of 3) Vol. II, are again connected to the appropriate channel selector proposition (terminals 7 and 8 for two channels per plug-in unit). The grid of one-half of this tube is also connected to a 100K potentiometer which shunts the terminals of the playback winding on the read-write head of the channel. A relatively small signal from the winding is sufficient to turn on this half of the tube (assuming that this channel has been selected and thus +125v has already been applied to the grid) since the cathodes are connected to +125v. A typical signal from the winding is about 0.6v in amplitude, as in the case of the recirculating register read windings, and this is amplified by the 5963 to a level of about 5v. Note that one plate of each channel's preamplifier is connected through a common external resistor of 10K (common to all the main memory preamplifiers) to the +225v regulated supply, while the second plate of each 5963 is tied directly to +225v.

The circuit of this preamplifier has been carefully designed to minimize the effects of external noise, and of shift in dc level, as indicated in the simplified schematic shown in Figure IV-29.

The right half of the 5963 acts as a cathode follower to stabilize the circuit. Thus, when the channel selector proposition goes true, a signal of +25v (the difference between the logical voltage levels of +100 and +125v) is applied to both grids. If the right triode were ab-

sent, this change would result in a large output signal. However, because this signal is applied to both grids, the cathode follower action of the right triode brings up both cathodes, and thus the tube is merely brought into the conducting region so that a signal of less than one volt from the read winding applied on the left grid will now be amplified in a linear manner by the left triode. Similarly, noise voltage or a shift in dc level at either the channel proposition input to both grids or at the input of the +225v regulated to both plates will effectively be cancelled out. On the other hand, the read signal from the winding is applied only to one grid, is amplified and produces an output signal at a much higher level than stray noise.

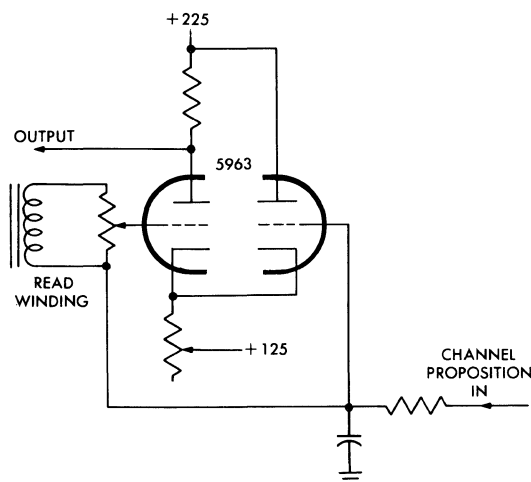


Figure IV-29. Simplified Schematic of a Main Memory Preamplifier

The signal from the memory preamplifier next is ac-coupled to the main memory limiter amplifier (20267337 shown in Figure 21 (3 of 3), Vol. II. Clamping diodes located externally on the Memory Clamp Chassis (see Figure 14, Vol. II) limit the input-level swing to +5v above and below 175v, that is, from a low level of +170 to a high of +180v. This limiting is done to eliminate excessive voltage swings, which could be in the order of 50v, re-

sulting from the switching transient or the record transient. In switching from one channel to another, for example, the output plate of the preceding 5963 preamplifier could go from +225 to +175 when the 125v supply is applied to the grid of that tube as its channel proposition goes true. If this 50v drop were applied to the coupling capacitor of the limiter amplifier, there would be an appreciable time required to discharge this capacitor before the limiter circuit would be sensitive to playback signals from the preamplifier. Similarly, in recording information on a main memory channel, a voltage as high as 50v might be induced in the read winding which is on the same head as the record winding. This record transient, too, would saturate the input of the limiter amplifier if the clamping diodes were not used.

The limiter amplifier itself, in the first stage, provides a gain of about 4.5 with a design that is exceptionally stable and unaffected by noise or voltage fluctuations on the plates of the tube. The fact that the grid of the second half of the dual triode is grounded means that the cathodes are relatively insensitive to changes in plate voltage until a signal is applied to the first grid. The signal output is amplified and limited because a positive signal at the first grid drives the second triode to cut off, while a negative signal cuts off the first triode. Thus the output waveform is squared on both half cycles and a signal amplitude of about 75v is applied at the first triode grid of the second 5687 stage.

Two outputs are available from the two plates (terminals 3 and 2), but only the output from terminal 2 is used, with an amplitude of about 30v peak and in phase with the input.

This output is then fed into a phase inverter (10667016), next into a clipper (10667017), and then into a logical driver (00467001), all of which have been

previously described. The outputs of this driver then trigger an unclamped flip-flop, as shown on Figure 14 Vol. II under "Memory Playback Circuits"; and the flip-flop outputs are finally inverted by a flip-flop driver to form M and M', the memory playback output propositions.

Note that at one bit-time during a word, P_0O_{13} , a clock product is formed and differentiated, then applied to grid terminal 10 of the unclamped flip-flop to turn it off so that a zero is always played back from the P_2O_{13} bit space of every memory cell. This is the final bit in each word. It should be noted that, at the end of a putaway into the main memory, the memory record circuitry turns off the memory record flip-flops (R_1 and R_2) at the end of P_1O_{13} (clock product with this pulse-time).

Thus, nothing ever can be recorded in the P_2O_{13} position of the main memory; and similar inhibiting prevents any playback from this pulse-time. At the start of the operating life of the computer, all the memory channels are filled with zeros. Hence the P_2O_{13} position always contains a zero in the main memory. This is done to avoid any "overshoot" of information from one cell to another, since a 1 at the P_2O_{13} position of a memory word might cause an incorrect indication at the P_0O_0 position of the next word if this deliberate blanking of P_2O_{13} were not assured.

The accompanying chart of waveforms (see Figure IV-30) shows what happens during both the recording and playback of the following four bits, 0 1 0 0, on any memory channel.

The word channel, M_w , and the synchronizing pulse M_s , are recorded by utilizing heads and circuitry identical with those used for the main memory channels. Then, after recording, the write

winding connected to the record circuit is usually disconnected. Reading M_w and M_s involves exactly the same circuit as described for the main memory. If it should be necessary to record one

of these again, for example, to renumber the word channel for minimum-access coding, then the record winding again may be connected while writing the new information on the channel.

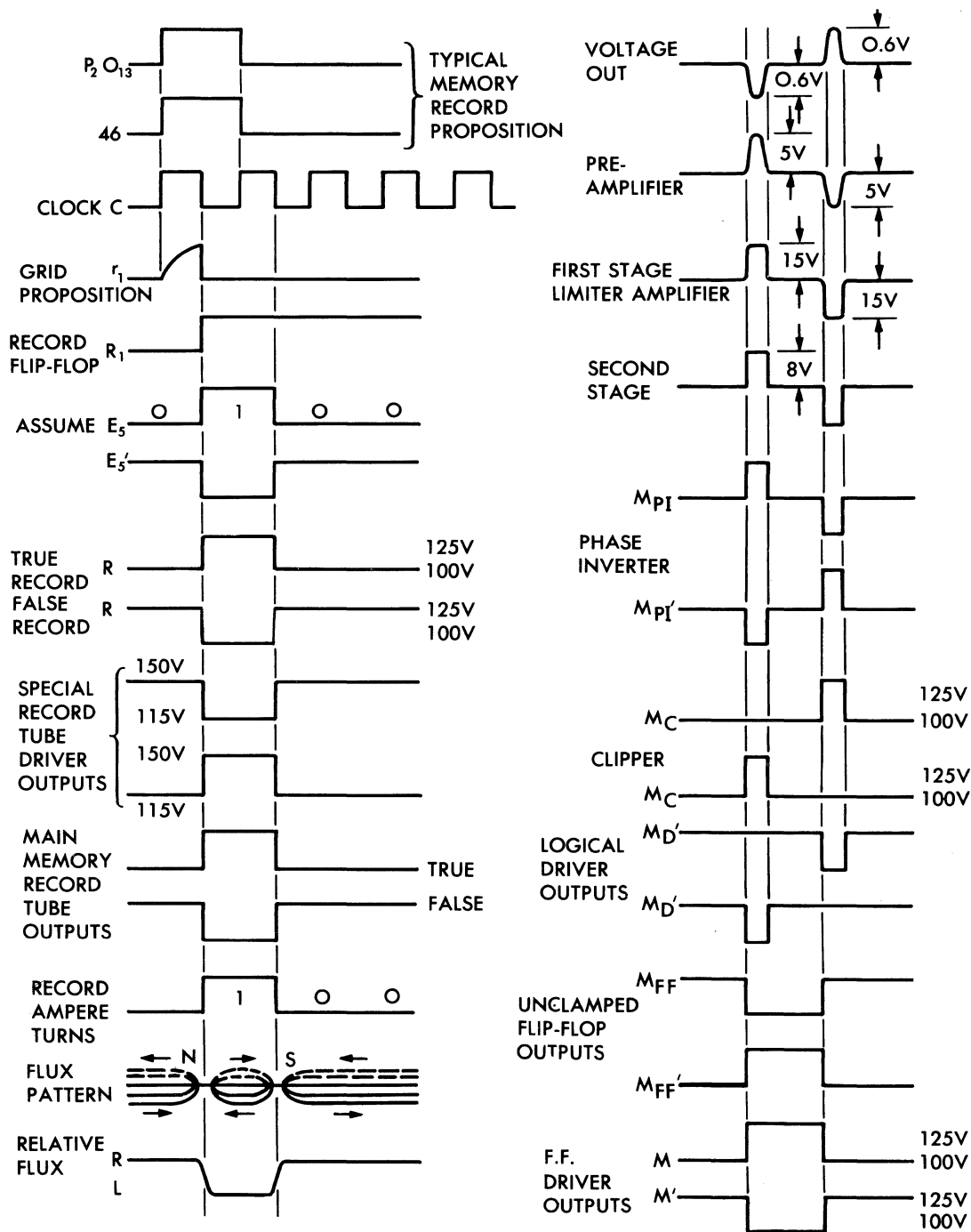


Figure IV-30. Idealized Waveforms of the Record and Playback Process in the Main Memory

Clock Circuit

The clock signal, which has been recorded on the memory drum is read from the drum by means of a playback winding on a head similar to those used for the main memory channels; and the signal from this winding is amplified in a preamplifier identical with the memory preamplifier. The gain control of this preamplifier (20267006), namely the 100K potentiometer across the winding, is used to adjust both the amplitude and the shape of the clock signal, since there are slight variations in the recording of the clock, see Figure 21(2 of 3) Vol. II. This adjustment should be made by company service personnel only.

As shown in Figure IV-31, the chart of the clock waveforms, the signal from the preamplifier goes to a Schmitt trigger. This Schmitt trigger (10667018) is a partial multivibrator with one stable state, see Figure 21(2 of 3) Vol. II.

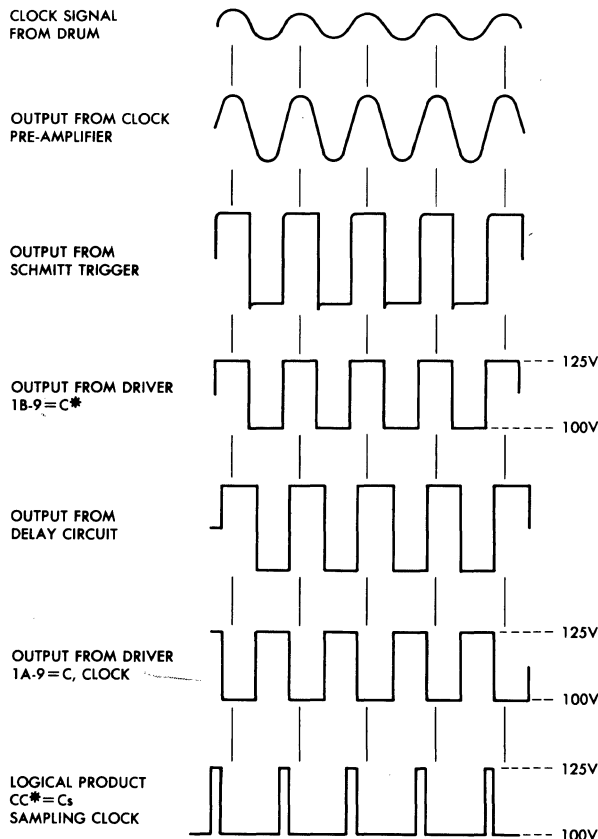
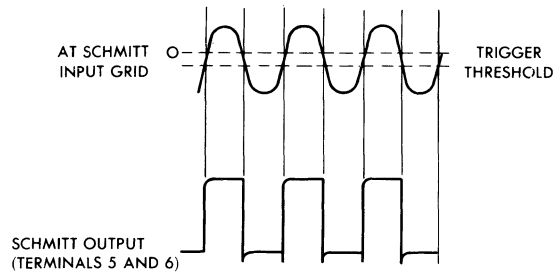


Figure IV-31. Idealized Clock Waveforms

The left half of the 12AT7 is normally conducting and is cut off by the pulse from the clock preamplifier. The top of the positive half of this pulse is partially clipped by the Schmitt input circuit as shown by the waveforms.

When the input level falls below a certain threshold voltage, the left half of the tube is cut off; and its plate potential rises, increasing the grid potential of the right half of the tube and also raising the voltage of both cathodes. Since the voltage on the left cathode increases the cut-off is rapid for the left half of the tube; and similarly, the right half switches on sharply, resulting in a sharp voltage drop on the right plate (true output on schematic). When the input pulse level again rises above the threshold of the circuit, the left half of the tube conducts; and its plate voltage drops. Voltages at both cathodes and at the grid of the right half also drop.



Again, the result is a rapid change in voltage at the output, the right plate, as shown in the waveforms above. Hence, the Schmitt trigger changes from one state to another in response to the input pulse, but both the rapid rise time and the rapid fall time of its output are independent of the shape of the input pulse, once the threshold voltage has been crossed in either direction.

Two identical outputs are taken from the Schmitt trigger, both in phase with the input from the clock preamplifier. One goes through a 1.2-microsecond delay circuit (20267004), which consists of an input cathode follower (half of the 5963 dual triode), then the delay line, and an output cathode follower (second half of

the 5963), see Figure 21(2 of 3) Vol. II. This delayed output is again split and part goes through a logical driver (006-67001), see Figure 21(1 of 3) Vol. II. Note that this is the special case previously mentioned in connection with this 5881 driver, where the input is ac-coupled to terminal 7, and the driver grid is connected to ground. The driver output, of course, is inverted and is clamped at the logical voltage levels of +125 and +100v. This output is the clock, C, used for timing all of the computer's operations.

The second output from the delay circuit is ac-coupled to a triode, the plate of which is connected to ground, and the cathode to -300v. The output from the plate of this triode, which has the regular clock waveform but negative voltage levels, is tied to the lower end of several logical sum resistors, (indicated by an asterisk on the schematics), as for example in the program counter sum networks: Figure 12(3 of 3), Vol. II. This is done so as to lower the logical sum rapidly to its correct false level of +100v when all of the terms go false. Distributed capacitance, prevents this action when the resistor is tied to ground. The device is used only when there are several terms in the logical sum.

The second output from the Schmitt trigger is applied to a logical driver, without intervening delay. This driver is connected so as to supply a true output (terminal 8) in phase with the input (terminal 3). This driver output, labeled C*, is then used to form a logical product with the clock, C, which has been delayed 1.2 microseconds and inverted. The product is a series of narrow pulses, 1.2 microseconds in duration, called the sampling clock, (C_s), used to

form a logical product with the clipper outputs of the read circuits of the recirculating registers. Note also the waveforms for C, C* and C_s in Figure IV-31.

The primary purpose of the sampling clock is elimination of the effect of the record transient from the read circuit output of the recirculating line. Because the write head and the read head of each register are physically so close (about 0.5 inch apart), the recording process induces a transient voltage in the read (playback) winding. This transient may be larger in amplitude than the playback voltage obtained by reading the desired information from the drum surface.

Eliminating the transient is a matter of timing. Note in the composite waveform chart for the E register (Figure IV-27) that E_w is delayed by one pulse-time after E_w and is triggered by the fall of the clock, C . Note also that the clipper outputs, E_c and E'_c , are in phase with the rise and fall of E_w , and hence also are synchronized with the clock. Meanwhile there is a finite delay interval of several microseconds required for generating the record transient in the read winding. As a result, the actual clipper outputs, sampling clock, and their products will appear somewhat as shown in the waveforms in Figure IV-32.

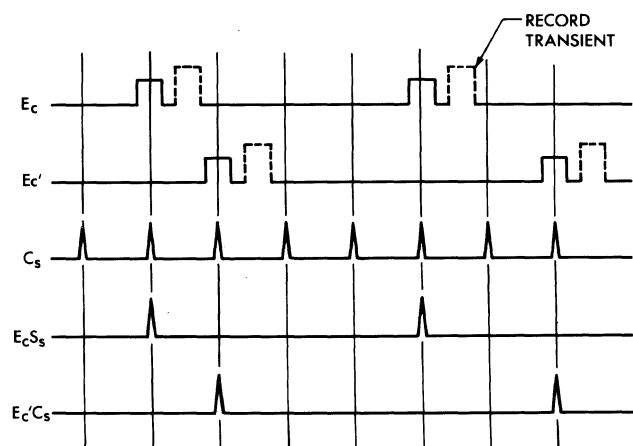


Figure IV-32. Idealized Waveforms of Clipper Outputs and Sampling Clock

Since the sampling clock, C_s , is synchronized exactly with the fall of the

clock, C , whereas the record transient is delayed by the read winding, the

products of the clipper outputs and C_s do not contain this transient.

LOGICAL STRUCTURE

Figure IV-33 is a block diagram showing the logical structure of the computer and indicating in simplified form how the computer operates. For example, suppose that a series of program commands are typed on the Flexowriter. Each time that a significant Flexowriter key is pressed, it sets the A_1 through A_5 flip-flops, as described in detail in this SECTION under "Flexowriter Circuitry and Operation". The logical diode networks transfer the information from these flip-flops to the memory, either main memory cells or the recirculating registers as called for in the logic. This logic, in turn, is controlled by the program counter flip-flops and the pulse and digit counters which act on the diode networks at the required times. After the computer has performed most of the operations called for in the program, the final commands may be to print the results and then half-computation. The results of computation are used to set up the A_{1-6} flip-flops, which actuate the appropriate Flexowriter keys and print the desired information.

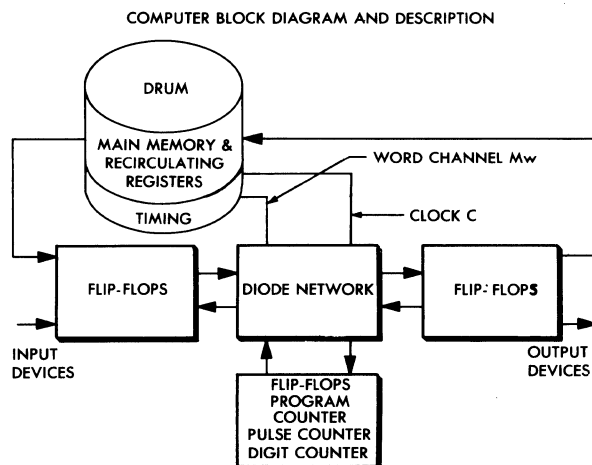


Figure IV-33. Logical Structure of the Computer

FLOW DIAGRAM

The time sequence of events for any particular program routine or cycle of computer operations, is shown by the flow diagram (Figure 1, Vol. II.) Each block of the flow diagram corresponds to a state of three-digit octal (nine-digit binary) counter, called the program counter. Any particular state of the program counter gates all the interconnections among the logical elements in the computer, hence the condition of the diode network; and thus it establishes the conditions required for the operation specified in the associated flow diagram block. Descriptions of the operations in all the blocks are provided in the Block Explanations in Volume II.

The program counter can change its state in going from one block to another, in either one of two ways as determined by the program counter logic: (1) it may count consecutively from one block to the next in numerical order, indicated on the flow diagram by horizontal lines between blocks; or (2) it may skip from one block to another, non-consecutive state, indicated by vertical lines between blocks. A special case of skipping is sticking, where the PC (program counter) is held in the same configuration and thus counts through the block again. The PC may skip or stick but not both.

Except when sticking, the PC stays in one block or one configuration for one word time, that is, 42 bits or about 420 microseconds. At the end of each word time, during the $P_{2^0 13}$ bit of each word, the state of the decision flip-flop, K , determines whether the PC is to count or skip. If K is false, the PC counts into the next block. If K is true, then the PC skips or sticks as shown in the

flow diagram. In some cases, as shown in the flow diagram, the counter can only skip to another block. The state of K at P_{2013} may be a result of the logical propositions being handled during that word time; or K may remain unchanged from a setting in a previous block.

The program counter flip-flops (N_1 through N_9) gate through the diode network the interconnection of all logical elements: such as, the appropriate flip-flops, recirculating lines, memory inputs and outputs, etc. If K is false at the end of a word time, the logic of these PC flip-flops cause them to operate as a nine-digit binary counter, advancing one bit at the end of the word. If K is true, these flip-flops are forced into a non-consecutive pattern (for skipping) determined by the previous PC number. Often, the state of only one N flip-flop need be changed to accomplish such a skip. For example, from PC24(000 010 100), K true causes a skip to PC64(000 110 100), requiring only that N_6 be set true (0 to 1).

<u>Operation</u>	<u>Result</u>
Turn on FILL switch on Computer Control Panel.	Connects Flexowriter to computer
Press CLEAR button	Clears and synchronizes computer circuits.
Type 1150 f	Fills 00 0000 11 50 0000 into E.
Type - (hyphen)	Transfers this number to H, putting address 1150 in m_2 of H.

Detailed Explanation: After the CLEAR button is pressed, the PC goes to Block 0 (rest). With the FILL switch on, striking "1150" on the Flexowriter does

the following for each octal digit: K is set false and PC counts to Block 1, where the signal sent from the Flexowriter is examined to determine whether it was a character (0 through 7) or a control symbol (d, o, s, f, hyphen, tab). Identification of the first digit "1" as a character sets K true; and the PC skips to Block 41 where A_1 , A_2 , and A_3 are inserted in the E line for one word time shifting E left one octal digit and inserting a 1 in the least significant position. (A_4 is inserted in the E line only when filling decimal.) K is always set true coming out of PC 41; the program counter skips to PC 40 where A_1 to A_5 are reset to 0; and K is set true for skip to PC 0, sticking there until the next key is struck. Thus, the sequence when "1150" is struck is as follows:

First condition of E	00 0000 0000 0000 (cleared)
Next condition of E	00 0000 0000 0001 (first 1 entered; A_1 true)
Next condition of E	00 0000 0000 0011 (A_1 true)
Next condition of E	00 0000 0000 0115 (A_1, A_3 true)
Final condition of E	00 0000 0000 1150 (all A flip-flops false)

Striking the "f" key fills in four octal zeros in the least significant digit positions of E. Now E contains 00 0000 1150 0000.

Striking the hyphen send the PC from Block 0 to Block 4, where K is set true for skip to PC 44. Here, the contents of E are transferred to H, and the PC skips to PC 40 and back to rest as before.

Now, continuing with the Extract Routine:

<u>Operation</u>		<u>Result</u>
Strike "s" key on Flexowriter.		Starts computation, which proceeds as follows:
Contents of registers at important points:		Computer finds m_2 of H in Blocks 6 and 7; reads (m_2) into H in PC 10, transfers former m_2 into G, adding 1 (1150 becomes 1151).
Entering	E 00 0000 0000 0000 F 00 0000 0000 0000	
PC 6:	G 00 0000 0000 0000 H 00 0000 1150 0000	
At end of PC 10	E 00 0000 0000 0000 F 00 0000 0000 0000 G 00 0000 1151 0000 H 32 0100 0101 0102	Looks up m_1 of H (0100) in PC 11 and PC 12; in PC 13 transfers (m_1) into E.
At end of PC 16	E 35 7236 3521 4263 F 00 2362 4125 3637 G 00 0000 1151 0000 H 32 0100 0101 0102	Looks up m_2 of H (0101) in PC 14 and PC 15; in PC 16 (m_2) goes into F.
<u>Contents of Registers</u>		<u>Result</u>
At end of PC 17	E 35 7236 3521 4263 F 00 2362 4125 3637 G 00 0000 1151 0000 H 32 0100 1151 0102	Address of next command (1151) is shifted from G into m_2 of H in PC 17. In Blocks 20 through 25, 1 is added octally to the Operation portion of H.
At end of PC 25	E 35 7236 3521 4263 F 00 2362 4125 3637 G 00 0000 1151 0000 H 00 0100 1151 0102	(32 in this example) in each block until in PC 25, this portion ($0_{13}0_{12}$) becomes 00 which sets K true for skip to PC 75.
At end of PC 75	E 35 7236 3521 4263 F 00 2362 4125 3637 G 00 0000 0000 0000 H 00 0100 1151 0102	In PC 75, the channel selector flip-flops, L_1 through L_5 , are again set up to select the channel of m_3 of H; the G line does not recirculate and hence is cleared.
At end of PC 77	E 35 7236 3521 4263 F 00 2362 4125 3627 G 25 2007 0674 2000 H 00 0100 1151 0102	In PC 76, the sector of m_3 of H is found; and in PC 77, (m_3) is transferred into G.
At end of PC 100	E 25 2227 0771 0223 F 00 0000 0000 0000 G 00 0000 0000 0000 H 00 0100 1151 0102	In PC 100, (E) is extracted into G, according to (F) using the equation developed on page IV-18, and then the result is fed back into E.

<u>Operation</u>	<u>Result</u>
At end of E 00 0000 0000 0000 PC 136 F 00 0000 0000 0000 G 00 0000 0000 0000 H 00 0100 1151 0102	F and G are cleared, since as the information is fed into E, the lines do not recirculate and hence the bits are erased. K is set true for skip to PC 134, where channel of m_3 of H is found; then count to PC 135, where sector of m_3 of H is found; count to PC 136, where (E) goes into this address, cell 0102 in this example; and (G) into E. Since the putaway to memory cell 0102 is a single word, A_{10} is left false, setting K true for skipback to PC 6, where the process of finding the next command (in 1151) begins.

Divide Routine

In the divide routine (considering its operation as "divide" with round off) the dividend is in the E register, and its magnitude is normally smaller than that of the divisor, contained in F. The quotient, which thus is less than 1, goes in G. If E is greater than F, an overflow occurs; and, unless this has been considered in programming, the computer will halt computation.

During the operation, the magnitudes of E and F are compared. If E is smaller than F, E is shifted left, since binary division begins at the MSBD; and a 0 is recorded in the quotient register (G) at the LSBD position. G is then shifted left for the next quotient pulse; F is not shifted. When E is larger than F, F is subtracted from E and the remainder is recorded in E, while a 1 is placed as the LSBD of G. Then both G and E are shifted left one bit. In either case (E smaller or larger than F), the shifted E is compared with F to determine whether the next operation should be a

subtraction or another shift.

To summarize, in the divide operation five computer functions are carried out simultaneously during one word time (Block 115):

1. Subtract F from E (if possible; that is, $E > F$).
2. Shift E left one bit.
3. Record a 1 or 0 in G. (1 if $E > F$ prior to shift).
4. Shift G left one bit.
5. Compare F with shifted E.

Next, consider the details of the "divide with round off" operation. Assume that the command is 23 030003010302 and is located in cell 0151; also that

(0300) - 00 4444 4444 4444

(0301) - 00 6666 6666 6666

It may be assumed also that the operations have proceeded exactly as in the previous example of Extract up to PC 17, whence they may be traced as follows:

<u>Contents of Registers</u>		<u>Result</u>
At end of PC 17	E 00 4444 4444 4444 F 00 6666 6666 6666 G 00 0000 0152 0000 H 23 0300 0152 0302	Address of next command is shifted into m_2 of H from G in PC 17.
At end of PC 34	E 00 4444 4444 4444 F 00 6666 6666 6666 G 00 0000 0152 0000 H 00 0300 0152 0302	In Blocks 20 through 34, 1 is added octally to the operation portion of H (23 for dr) in each Block until in PC 34, this portion (o_{12-13}) of H becomes 00, setting K true for skip to PC 114.
At end of PC 114	E 00 4444 4444 4444 F 00 6666 6666 6666 G 00 0000 0000 0010 H 00 0300 0152 0302 (using binary numbers below for clarity)	In PC 114, E and F are compared with A_1 set true if $E > F$, false if $E > F$. The sign of E is stored in A_4 , of F in A_6 . A_7 is set false if $E > F$ to indicate overflow. A marker pulse is placed in
At end of first word time in PC 115	E 000001 001 001..001 000 F 000000 110 001..110 001 G 000000 000 000..010 000 H 000000 000 011..000 010	G at P_0^0 , meanwhile clearing G of its previous contents. K is set false for count to PC 115. In PC 115, using this numerical
During second word time in PC 115	E 000000 010 010..010 010 F 000000 110 110..110 110 G 000000 000 000..010 001 H 000000 000 011..000 010	example, A_1 is true, hence $E > F$. Flip-flop A_8 is used to make the comparison between (E) and (F). Flip-flop A_1 is then set
At end of second word time of PC 115	E 000000 100 100..100 100 F 000000 110 110..110 110 G 000000 000 000..000 010 H 000000 000 011..000 010	from A_8 to determine whether F is to be subtracted from E or not. Thus, F is not subtracted from E when A_1 is true; and a
At end of third word time of PC 115	E 000001 001 001..001 000 F 000000 110 110..110 110 G 000000 000 000..000 100 H 000000 000 011..000 010	0 is placed as the LSBD of G.
At end of 36th word time of PC 115	E 000001 000 000..000 000 F 000000 110 110..110 110 G 001000 010 010..101 010 H 000000 000 011..000 010	Now, shift E and G left one bit; again compare E and F, using A_8 , and if A_1 is set false from A_8 ($E > F$), subtract F from E, and record a 1 as LSBD of G. Now shift
(Note $E > F$ after final shift, hence A_1 set false. G contains quotient before round-off.)		remainder, which is in E, left one bit again and compare with F. The process continues as above (sticking in Block 115) until the marker placed in G in PC 114 at the
At end of PC 117	E 000000 101 010..101 011 F 000000 110 110..110 110 G 000000 000 000..000 000 H 000000 000 011..000 010	P_0^0 position has been shifted to $P_0^{0_{13}}$.

Contents of Registers

(Note E now contains quotient, after round-off, with 1 added to LSBD using A_1 to set A_3 true and A_3 to add the 1 in PC 117.)

Results

or 36 bit times -- equivalent to the magnitude of the 36-bit numbers in E and F.

At the end of the 36 shifts of E, a final comparison is made with F and this determines the round-off information for the quotient. A_1 is set false from A_8 if $E > F$ as shown at left for end of 36th word time. In PC 116, the sign of the quotient is put in G, using A_4 , which held the sign of the dividend, and A_6 which held the sign of the divisor. If A_{10} is false (divide and round-off), set K false for count to PC 117, where 1 is added to G if $E > F$ is final comparison (as above); and then (G) is transferred to E.

In PC 117, K is set true for a skip PC 134, and from here through PC 136 and back to PC 6 for the next command, the operations are the same as in the previous example of the Extract Routine.

INPUT-OUTPUT ROUTINES AND CIRCUITS

FLEXOWRITER CIRCUITRY AND OPERATION

Input to Computer

The flexowriter is the primary input-output device for the computer. All information filled into the computer from the Flexowriter enters the 42 binary digit E recirculating register. The sequence and electronic circuitry involved in this transfer operation are described in the following paragraphs.

When a Flexowriter key is struck, it operates a mechanical matrix which closes the appropriate contacts in a group of

six lines to the computer; and voltage is applied from the power supply of the Flexowriter along those lines affected. Thus, the output of the Flexowriter is essentially a six-digit parallel binary code, expressed as the presence or absence of voltage on each of the six lines to the computer. TABLE IV-9 lists the 20 significant Flexowriter keys and their coded representations on the six output lines to the computer (numbered 6, 5, 4, 3, 2, 1, in order of decreasing significance).

These output lines from the Flexowriter are connected to six fill relays in the

Flex. Key Char- acters	Computer Flip-Flop State $A_5 A_4 A_3 A_2 A_1$	Octal Flex. Nos. $F_X \#$	Binary Configuration of Flex. Nos. $6 \ 1 \ 2 \ 3 \ 4 \ 5$ $43 \ 48 \ 47 \ 46 \ 45 \ 44$ $6 \ 5 \ 4 \ 3 \ 2 \ 1$	Explanation
				Flex. Output Lines
				Relay Nos. in Computer
				Relay Proposition Nos.
0	0 0 0 0 0	37	0 1 1 1 1 1	for octal mode filling
1	0 0 0 0 1	11	0 0 1 0 0 1	
2	0 0 0 1 0	74	1 1 1 1 0 0	
3	0 0 0 1 1	70	1 1 1 0 0 0	
4	0 0 1 0 0	64	1 1 0 1 0 0	for decimal mode filling
5	0 0 1 0 1	62	1 1 0 0 1 0	
6	0 0 1 1 0	66	1 1 0 1 1 0	
7	0 0 1 1 1	72	1 1 1 0 1 0	
8	0 1 0 0 0	60	1 1 0 0 0 0	Same as 0, either mode
9	0 1 0 0 1	33	0 1 1 0 1 1	
+	0 0 0 0 0	52	1 0 1 0 1 0	
-	0 0 0 1 0	56	1 0 1 1 1 0	
Neg. Space	0 1 1 0 0	04	0 0 0 1 0 0	Special characters for decimal mode use.
period	0 1 1 1 1	42	1 0 0 0 1 0	In octal mode, space fills as 4, period as 7.
Control Symbols				
f	1 0 0 0 0	26	0 1 0 1 1 0	"Fill four zero's"
d	1 1 0 0 1	22	0 1 0 0 1 0	Clear input register and fill decimal mode.
o	1 1 0 0 0	03	0 0 0 0 1 1	Clear input register and fill octal mode.
- (hyphen)	1 1 1 0 0	50	1 0 1 0 0 0	Transfer contents of input register to control reg.
tab	1 1 1 1 0	51	1 0 1 0 0 1	Transfer contents of input register to memory
s	1 1 1 1 1	24	0 1 0 1 0 0	Start computation

TABLE IV-9. Flexowriter - to - Computer Input Settings

computer, correspondingly numbered 43, 48, 47, 46, 45, 44. Whenever a Flexowriter line has voltage on it, the corresponding computer relay is energized and a true signal (+125v) is sent to a logical matrix. This logical voltage is supplied by the computer. The 90v from the Flexowriter merely operates the relays. Any relay not energized sends a false signal (+100 v) to the matrix. (See Figure 27, Vol. II.) These 20 matrix outputs are then combined into logical sums, called Flexowriter Sums (FXS). There are five of these sum circuits, designated as $FXSa_{1-5}$ (see Figure 14, Vol. II); and each is used as a grid proposition to turn on the correspondingly numbered flip-flops, A_1 through A_5 . The setting of these five flip-flops conveys the required information, initiated by striking a Flexowriter key, as shown in TABLE IV-9.

In TABLE IV-9, the numbers "6 1 2 3 4 5" under "Binary Configuration of Flexowriter Numbers" corresponds to the numbering of positions on the punched Flexowriter tape, and also to the six lines from the Flexowriter to the computer.

The six relay propositions and their primes are developed by the operation of relays, as indicated in Figure IV-34.

The six relay propositions and their primes provide the 12 inputs to the logical matrix, and the 20 outputs of this matrix are added logically to form the Flexowriter Sums.

To illustrate one complete Flexowriter input operation, suppose the "3" key on the Flexowriter to be struck. This puts voltage on Flexowriter output lines 6, 1 and 2, which energize relays 43, 48, and 47 in the computer, as shown in TABLE IV-9. These relays cause relay propositions 6, 5, and 4 to go true, or rise to +125v; while relay propositions 3, 2, and 1 remain false at +100v. With these six inputs to the logical matrix, the out-

puts combine to make the two sums $FXSa_1$ and $FXSa_2$ go true. Thus, flip-flops A_1 and A_2 are turned on.

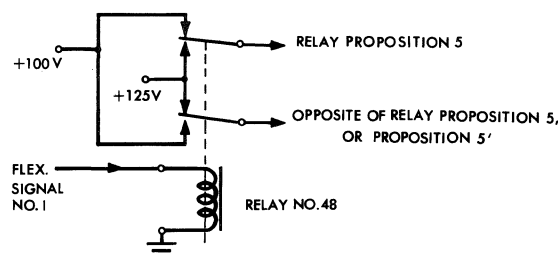


Figure IV-34. Operation of a Fill Relay in the Computer

As indicated in TABLE IV-9, the octal Flexowriter number rather than the key symbol is used in the expressions for Flexowriter Sums. Thus $FXSa_1 = (11+70+62+72+33+42+22+24) + S$ means that any of the Flexowriter symbols 1, 3, 5, 7, 9, period, d, or s will turn on flip-flop A_1 . The capital S in a circle at the end of the sum should not be confused with the small s, represented by Flexowriter octal number 24. This encircled capital S is the symbol for the "start switch proposition" which results from pressing the COMPUTE button on the Computer Control Panel. However, striking the s key on the Flexowriter or pressing this COMPUTE button will produce the same final effect in the computer; namely, turning on flip-flops A_1 through A_5 .

To carry the input from the Flexowriter further, it is necessary to consult the Block Explanations in Volume II. Note that in Block 0, striking any of the 20 significant Flexowriter keys sets the A_1 to A_5 flip-flops and also sets K false to count to PC 1. In this block, the state of A_5 is examined to determine whether to count to Block 2 or skip to Block 41. A_5 is true for control symbols (f, d, etc. see TABLE IV-9) and false for characters (0, 1, 2, etc.). Thus, a character such as 3 which was used previously in the example, will make the

program counter skip to PC 41, where the contents of flip-flops A_1 to A_3 (octal numbers) or A_1 to A_4 (decimal) are shifted into the LSBD positions of the E register. Next, the PC skips to PC 40, where all of the A flip-flops are turned off, and then back to PC 0 to await the next Flexowriter signal.

As shown in the Block Explanations and Flow Diagram, A_5 true in PC 1 would mean a control symbol and would cause a count to PC 2, where the state of A_4 is examined. Consulting TABLE IV-9, it is seen that A_5 is true (1) and A_4 false (0) only when the f key is struck. Now the PC skips from PC 2 to PC 54 and then counts through PC 57. Blocks 54-57 each shift the contents of E left one octal digit. This normally results in filling four octal zeros at the right (least significant) end of the E register, and shifting any previously-filled digits four octal places left. From PC 57, this operation calls for skipping to PC 40 and thence to PC 0, as before.

For any control symbol other than f, the PC counts to PC 3, where A_3 is tested.

A_3 is false only if the symbol was d or o. In such case, the PC skips to PC 46, where A_6 is turned on if A_1 is true, and off if A_1 is false. This setting of A_6 determines whether, in subsequent filling of characters (numbers), the filling is done octally using the contents of A_1 to A_3 to shift bits into the E line (A_6 false) or decimally using the contents of A_1 to A_4 (A_6 true). Thus, striking the Flexowriter o key causes octal mode filling until such time as the d key is pressed, when decimal filling ensues.

In PC 4, A_2 is tested. If A_2 is false, the Flexowriter symbol is identified as a hyphen and the PC skips to PC 44 where all information previously filled in the input register, E, is transferred to the control register, H. From PC

44, the skip is to PC 40 and back to PC 0. A_2 true results in counting from PC 4 to PC 5.

A_1 is examined in PC 5. If it is false, the symbol is a tab, and the PC skips to PC 45 where a memory channel is selected. It then counts to PC 46 where the sector is found, counts to PC 47 where the information from E is transferred to the selected memory cell, counts to PC 50, where 1 is added to LSOD in H to provide the next putaway address, and finally skips to PC 40 and PC 0.

If all the A_1 to A_5 flip-flops are turned on (true), the PC into BLOCK 6, the first step in all computation routines, as explained previously.

This condition ($A_1 - A_5$) corresponds to striking the Flexowriter s key. The COMPUTE button on the control panel also turns on the five A flip-flops and causes the PC to count to PC 1 and thence to PC 6. Thus, the effect of this button is the same as that of the s key.

The first step in all of these filling operations by the 102-A Computer, namely, the count from PC 0 (rest) to PC 1, is initiated by the start proposition, S.

The start proposition is the logical sum of all 20 Flexowriter conversion matrix outputs plus the "start switch proposition" S, as indicated in Figure 14, Vol. II. Thus, the start proposition goes true when any Flexowriter character or control key is struck or when the COMPUTE switch is closed.

The additional circuitry shown in the Start Circuit is required to eliminate the undesirable effects of contact bounce and unequal operating times of relays.

The upper part of the start circuit shows that any of the six relay propositions (indicated as 1 through 6) from the Flexowriter, or the start proposition S, will initiate a positive signal to the

logical driver. This signal, as shown on the waveform chart, Figure IV-35 has a duration of about 50 ms, when initiated by any of the relay propositions. The Fill switch, when thrown to the non-Fill position, holds this signal at a high level and inhibits the action of the start circuit.

The output from the logical driver is inverted; and as shown in the schematic (Figure 14, Vol. II) this negative signal goes to two points. One is the diode 1 (54DD), where this signal acts to keep a logical product from becoming true until the proper time. The other destination of this signal is the first one-shot, which delivers a negative signal having a duration of about 20 ms. The positive-going trailing edge of this signal triggers a second one-shot delay circuit, in this case with a period of about 3 ms, which generates a sampling proposition approximately at the center of the relay signal and applies it to diode 1 (52DD) and to flip-flops A_1 to A_5 . This diode forms part of a logical product network with diode 1 (51DD).

The input to this latter diode, 1 (51DD), comes through a cathode follower from a logical sum consisting of the four Flexowriter sums, FXS_{a_1} through FXS_{a_4} , plus Flexowriter numbers 37, 26, and 52. This means that FXS_{a_5} is included since all of its terms except 26 are contained in the other four Flexowriter sums. Hence, there is a true input to diode 1 (51DD) if any significant key is pressed. Thus, pressing a significant key will make the logical product of 1 (51DD) and 1 (52DD) go true. This product will have a duration of about 3 ms, corresponding to the period of the second one-shot described previously. The trailing edge of this positive pulse triggers the start flip-flop (plug-in unit TC1-C15), as shown in Figure 14, Vol. II; and now a "true" signal is applied to diode 1 (53DD), which is part of the logical product circuit with 1 (54DD) receiving the driver out-

put. As shown on the waveform chart, this $E_1 E_d$ product goes true after the original relay pulse has gone false.

This product, going to diode 4 (150), is the grid proposition o_k which forms a clock product to set the K flip-flop false. When this happens, the clock product $K'C$ turns off the start flip-flop and the circuit is ready for another start proposition.

The special features of this start circuit include an adequate delay so that the electrical effects of any chatter or noise from the Flexowriter relays will have disappeared before the start proposition can go true: e.g., logical product 1 (53DD) and 1 (54DD) true. Furthermore, a significant Flexowriter key must be struck before the start circuit will operate. In addition, the entire timing sequence is carefully planned so that only a correct input signal will make the start proposition true.

Output to Flexowriter PRINT OUT

A primary method of taking information from the computer in useful form is to print out the results of computation on the Flexowriter. Programming the print command has been described previously. Further details are provided in the Block Explanation Forms for Blocks 36 and 204-276 (see Vol. II.). It is helpful also to study the Flexowriter instruction manual covering normal operation of this machine.

One feature of the electronic circuitry in connection with the print command is illustrated in Figure 25(2 of 2), Vol. II. Note the Go Ahead relay, which is in Upper Tube Chassis No. 2 (UTC-2) of the computer.

This Go Ahead relay normally is turned on, which means that there is power to make the Go Ahead proposition in the computer go true and stay true, and as a result the computer can proceed with its computation without waiting for the Flexowriter, as it must do during the

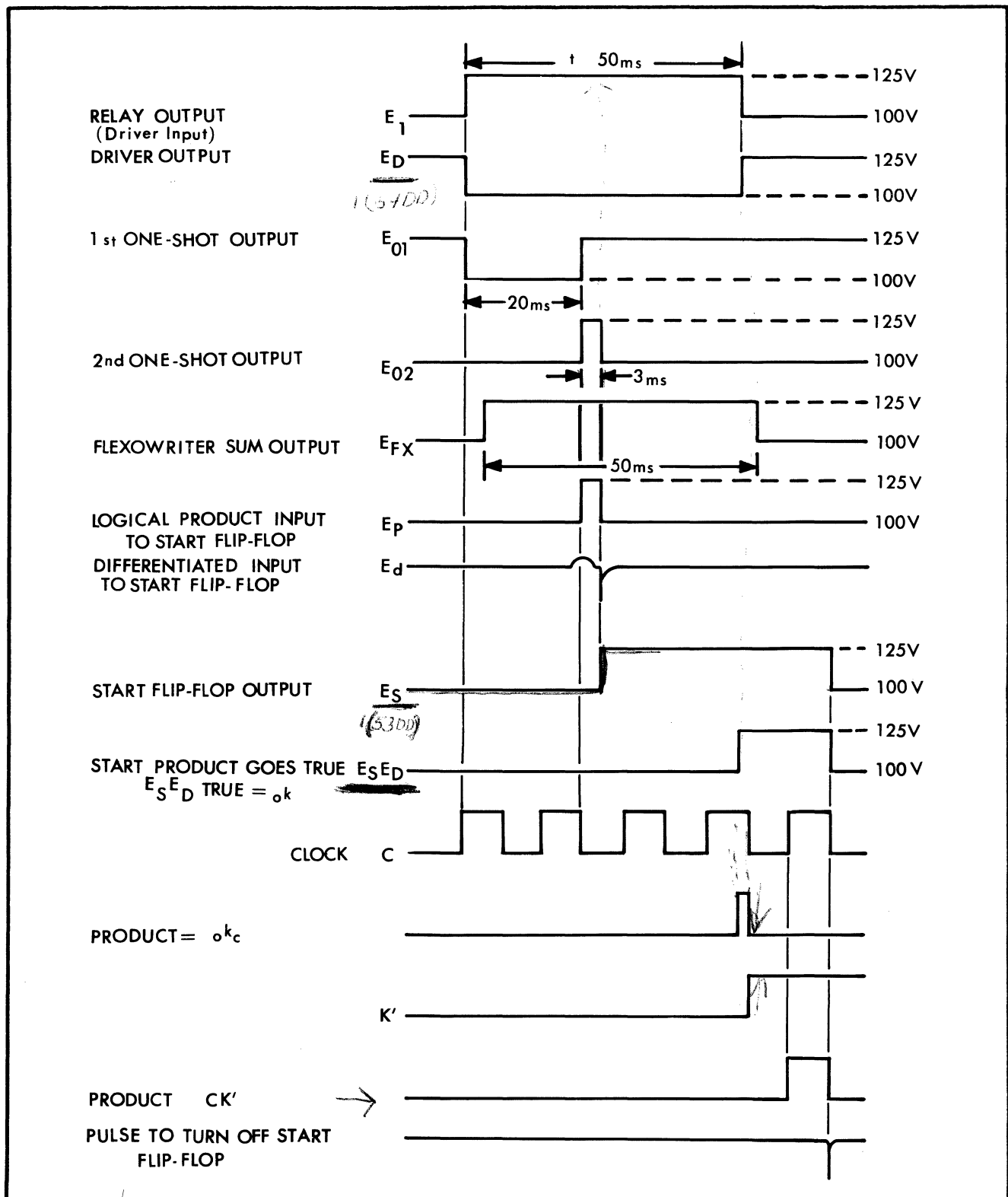


Figure IV-35. Idealized Waveforms in the Start Circuit

print routine.

During all blocks of the print routine (PC's 204-276), print relay, No. 33 in the computer is turned on so that contacts 2 and 3 are closed. On the other hand, the reader control relay in the Flexowriter is always off during the print routine: it is turned on only when the Flexowriter is being used to read paper tape to fill the computer.

Before every block in the flow diagram labeled "Actuate Key" (see Figure 1, Vol. II) as for example in Block 204, flip-flops A_1 through A_6 are setup correctly to actuate relays 37 through 42 in the computer. These relays, in turn, actuate the appropriate translator decoding magnets in the Flexowriter (TD M1 through TDM6) to setup the desired key for printing.

During a typical "Actuate Key" block (for example, Block 206), print cycle relay No. 28 (Figure 26, Vol. II) is turned on and contacts 2 and 3 close. The translator clutch release magnet is actuated, permitting the translator shaft to turn and open the cycle contact in the Go Ahead relay. This turns off the Go Ahead relay and the Go Ahead proposition in the computer goes false. As a result, the K flip-flop remains false so that the program counter counts to the next block, "Wait for Flex" (Block 207).

In such a typical "Wait for Flex" block, the program counter sticks until the actuated Flexowriter key has completed its printing operation. Each time the PC sticks, K is set true; it remains true until the cycle contact closes again (contacts 2 and 3) and the Go Ahead relay is turned on. This makes the Go Ahead proposition true and sets K false for a count to the next block.

It should be noted that the automatic carriage return relay contact No. 1 opens the circuit to the Go Ahead relay rapidly, and thus quickly turns this relay off when necessary. For instance,

it opens the current path to the Go Ahead relay when the computer tends to print more rapidly than the cycle contact will permit. The same automatic carriage return contact No. 1 stops the Flexowriter reader, when it is reading tape, at the correct point on the tape until the carriage of the typewriter returns.

If the computer sends any of the three delay characters (carriage return, tab, and backspace) to the Flexowriter, then delay relay No. 32 is closed by proposition $A_2'A_4A_6'$. The cycle contact (3-1) closes so that power goes through delay relay No. 32 and print relay No. 33 to the delay control relay in the Flexowriter, which now is turned on and electrically "latched" on. This latter relay now withholds power from the Go Ahead relay even when the cycle contact returns to position 3-2. The delay control relay remains on until one of the delay character (carriage return, tab and backspace) contacts opens. These delay characters require a longer Flexowriter operating time. Hence, the need arises for the delay control relay to keep the Go Ahead relay turned off while these characters are effective.

In general, when the computer is causing it to print, the Flexowriter performs as if it were reading paper tape. Functions of the tape reader are duplicated by relays in the computer.

The relay circuits which handle the information flow between the computer and Flexowriter are shown schematically in Figure 26, Vol. II. The process is described in TABLE IV-10.

Observe that the logical product of the appropriate flip-flop (A_1 to A_6) and the required program counter number triggers a relay driver which energizes one of the six computer relays. These relays, in turn, are connected to the Flexowriter magnets which operate the appropriate keys to print out the desired information. TABLE IV-11 shows the computer - to - Flexowriter conversion code.

In Computer					In Flexowriter
Prop.	Driver Location	Grid Pin No.	Plate Pin No.	Relay No.	Flexowriter Magnet
A ₆ '	TCA4	8	5	37	6
A ₅ '	TCB4	9	3	42	1
A ₃ '	TCB4	8	5	41	2
A ₁ '	TCA4	10	11	40	3
A ₂	TCA4	7	6	39	4
A ₄	TCA4	9	3	38	5

TABLE IV-10. Information Flow Between Computer and Flexowriter.

Notes for TABLE IV-11:

1. Characters having the binary configuration shown in Column 1 are logically converted in the decimal or octal routine to the configuration shown in Column 2.
2. In the case of alphabetic mode, the A flip-flops are set-up directly into the Column 2 configuration without the conversion as explained above.
3. Column 3 is the octal representation of Column 2.
4. Column 4 is the same as Column 2 but is rearranged to indicate which propositions actuate the Flexowriter bars. The Flexowriter code numbers (6, 1, 2, 3, 4, 5) correspond to the hole code of the tape, from left to right as viewed from the operator's position, and also the numbers of the magnets and bars. In punching tape, a binary 1 in any position of column 4 causes a perforation in that position on the tape.

When the print command causes the computer to type out data automatically, the format of a typical command is as follows: 21 1501 0103 0002. This command means: "Print (octal code 21) the contents of cell 1501 (m_1) according to the printing mode (octal, decimal, or alphabetic) contained in cell 0103 (m_2). Since the octal number in m_3 is 2, type out two words, starting with (1501) and continuing with the next word in order, (1502)."

Assume (1501) = 35 1651 1652 1650; (1502) = 36 1001 1002 1003;

and (0103) = 02 0000 0000 0010 = (m_2). The contents of m_2 mean: "Type the octal address of the cell being printed followed by a space, and then the contents of the cell, including sign digits and magnitude, as an octal number (specified by sign digits 02 in m_2). The number of digits to be printed from the magnitude sections of cells 1501 and 1502 is 11 (controlled by the position of the binary 1 in the magnitude section of m_2)."

Flex magnets #2 & holes in type

Column 1 Binary Configuration <i>P240</i>	Column 2 Configuration After Conversion <i>P241</i>	Column 3 Octal Code	Column 4 <i>same as type computer code</i> A ₆ ' A ₅ ' A ₃ ' A ₁ ' A ₂ ' A ₄	Column 5 Flexewriter Character Printed
A ₄ A ₃ A ₂ A ₁	A ₆ A ₅ A ₄ A ₃ A ₂ A ₁		6' 1 2 3 4 5	Upper Lower
	1 0 0 0 0 1	41	0 1 1 0 0 0	A a
	1 0 1 1 1 1	57	0 1 0 0 1 1	B b
	1 1 0 0 1 0	62	0 0 1 1 1 0	C c
	1 0 0 1 1 1	47	0 1 0 0 1 0	D d
	1 0 0 1 0 1	45	0 1 0 0 0 0	E e
	1 0 0 1 1 0	46	0 1 0 1 1 0	F f
	1 1 1 0 1 1	73	0 0 1 0 1 1	G g
	1 1 1 1 0 0	74	0 0 0 1 0 1	H h
	1 1 0 0 0 0	60	0 0 1 1 0 0	I i
	1 0 0 0 1 1	43	0 1 1 0 1 1	J j
	1 0 0 0 1 0	42	0 1 1 1 1 0	K k
0 0 0 1	1 1 1 0 0 1	71	0 0 1 0 0 1	L l
	1 1 1 1 1 0	76	0 0 0 1 1 1	M m
0 0 1 1	1 1 0 1 1 0	66	0 0 0 1 1 0	N n
	1 1 1 1 1 1	77	0 0 0 0 1 1	O o
0 0 0 1	1 1 1 0 0 0	70	0 0 1 1 0 1	P p
	1 0 1 0 0 0	50	0 1 1 1 0 1	Q q
	1 1 0 0 1 1	63	0 0 1 0 1 0	R r
	1 0 0 1 0 0	44	0 1 0 1 0 0	S s
	1 1 1 1 0 1	75	0 0 0 0 0 1	T t
	1 0 0 0 0 0	40	0 1 1 1 0 0	U u
	1 1 1 0 0 1	72	0 0 1 1 1 1	V v
	1 0 1 0 0 1	51	0 1 1 0 0 1	W w
	1 0 1 1 1 0	56	0 1 0 1 1 1	X x
	1 0 1 1 0 0	54	0 1 0 1 0 1	Y y
	1 0 1 1 0 1	55	0 1 0 0 0 1	Z z
0 0 0 0	1 0 1 0 1 0	52	0 1 1 1 1 1	0)
0 0 1 0	0 0 0 0 0 0	00	1 1 1 1 0 0	2 &
0 0 1 1	0 0 0 0 0 1	01	1 1 1 0 0 0	3 /
0 1 0 0	0 0 0 0 1 0	04	1 1 0 1 0 0	4 \$
0 1 0 1	0 0 0 0 1 1	07	1 1 0 0 1 0	5 %
0 1 1 0	0 0 0 0 1 0	06	1 1 0 1 1 0	6 ?
0 1 1 1	0 0 0 0 1 1	03	1 1 1 0 1 0	7 !
1 0 0 0	0 0 0 0 1 0	05	1 1 0 0 0 0	8 *
1 0 0 1	1 0 1 0 1 1	53	0 1 1 0 1 1	9 (
0 0 1 0	0 1 0 0 1 0	22	1 0 1 1 1 0	- (minus) ¢
0 0 0 0	0 1 0 0 1 1	23	1 0 1 0 1 0	+ ,
0 0 0 1	0 1 0 0 0 0	20	1 0 1 1 0 0	, " :
	0 1 0 1 1 0	26	1 0 0 1 1 0	, . :
1 1 1 1	0 1 0 1 1 1	27	1 0 0 0 1 0	- (hyphen) _
	0 1 0 0 0 1	21	1 0 1 0 0 0	;
	0 1 0 1 0 0	24	1 0 0 1 0 0	Lower Case
	0 1 1 0 1 0	32	1 0 1 1 1 1	Upper Case
	0 1 1 1 1 0	36	1 0 0 1 1 1	Back Space
	0 0 1 1 0 1	15	1 1 0 0 0 1	Space Bar
1 1 0 0	1 1 0 1 0 0	64	0 0 0 1 0 0	Code Delete
	0 0 1 0 1 0	12	1 1 1 1 1 1	Stop Code
	0 1 1 1 1 1	37	1 0 0 0 1 1	Color Shift
	1 1 0 1 1 1	67	0 0 0 0 1 0	Tab
	0 1 1 0 0 1	31	1 0 1 0 0 1	Carriage Return
	0 1 1 1 0 0	34	1 0 0 1 0 1	

TABLE IV-11. Computer-to-Flexewriter Conversion Chart

SEE NOTES ON OTHER SIDE

<u>Step</u>	<u>Result</u>
(Assume computer and Flexowriter are turned on; the CLEAR button has been pressed. Assume this print command given in the preceding paragraph is in cell 0120.)	
Type 0120 F	Fills 00 0000 0120 0000 into E Transfers this
Type - (hyphen)	number to H, putting address 0120 in m_2 of H.
Strike "s" key on Flexowriter	Starts computer operation, which proceeds as
At end of E 35 1651 1652 1650	follows: Computer finds m_2 of H in PC's 6 and
PC 16: F 02 0000 0000 0010	7; reads (m_2) into H in PC 10; looks up m_1 of H
G 00 0000 0121 0000	(1501) in PC's 11 and 12, and in PC 13 transfers
H 21 1501 0103 0002	(m_1) into E. Finds m_2 of H (0103) in PC's 14 and
	15, and transfers (m_2) into F in PC 16. Address
	of next command is shifted from G into m_2 of H
	in PC 17.
	In Blocks 20 through 36, 1 is added each time to
	the function portion of H, which started at 21
	(pr). In Block 36, octal 21 becomes 00, K is set
	true, and PC skips to PC 236 for start of print
	routine. Here G is cleared. In PC 236, P_1^0 of F
	is tested to ascertain whether addresses and
	numbers, or numbers only are to be printed,
	according to the print-out code.
At end of E 00 1501 0121 0002	Since in this example P_1^0 contains a 1, K is
PC 237 F 02 0000 0000 0010	set false for count to PC 237, where a 1 is placed
G 00 1000 0000 0000	in P_0^0 of G, and (H) goes into E. This is done
H 00 1501 0121 0002	to provide the address for print-out in m_1 of E.
	Count to PC 240. Here, flip-flops A_1 - A_4 are
	set up from (E), using the marker in G, and flip-
	flops A_7 and A_8 which follow that marker, K is
	set false for count to PC 241 where A_1 - A_6 are
	set in Flexowriter code from A_1 - A_4 . In PC 242,
	the proper Flexowriter key is struck (in this
	case 1, the first digit of the address 1501 from
	E) as previously set up in A_1 - A_6 . Stick in PC
	242 until the "Go Ahead" signal from the Flexo-

<u>Step</u>		<u>Result</u>
		writer goes false. Then count to PC 243 and wait for Flex "Go Ahead" signal to go true, meanwhile resetting A_5 and A_6 to zero. In PC 244, there is a test for the last digit of the address, with K remaining false unless there is a marker in 0_8 of G. Hence, count to PC 245, where (G) is shifted one octal digit in the right, and K is set true for skip back to PC 240.
At end of	E 00 1501 0103 0002	The process is repeated through Block 240-245 until after three shifts of (G), the Flexowriter has printed the address 1501 as called for by the code in F. At this point, the test in PC 244 sets K true for skip to PC 204, where A_1 - A_6 are set for "space" and K is set false. In this block, G does not recirculate and accordingly clears itself. In PC's 205-207, the usual routine is followed to actuate the Flexowriter to indicate a space.
PC 245	F 02 0000 0000 0010	
(first	G 00 0100 0000 0000	
time)	H 00 1501 0121 0002	
At end of	E 00 1501 0103 0002	In PC's 210 and 211, the channel and sector of m_1 of H (in this case, 1501) are found, and in PC 212 the contents of this cell are read from the memory into E. In the same block, $P_0^{0_{13}}$ of F is tested for alphabetical print. In this example, the command specifies octal print by the code in F, and K stays false for count to PC 213 where $P_0^{0_{12}}$ of F is tested for octal or decimal. Again, in this example, K stays false, and count to PC 214, where a marker 1 is placed in $P_0^{0_{13}}$ of G to select the octal digit to be printed.
PC 245	F 02 0000 0000 0010	
(third	G 00 0001 0000 0000	
time)	H 00 1501 0121 0002	
At end of	E 35 1651 1652 1650	In PC 215, A_1 - A_3 are set up from (E) using the marker in G and A_7 , A_8 which follow that marker. A_4 is set 0 because the number is octal. In PC's 216 through 222 the action is similar to that
PC 212	F 02 0000 0000 0010	
	G 00 0000 0000 0000	
	H 00 1501 0121 0002	
At end of	E 35 1651 1652 1650	
PC 215	F 02 0000 0000 0010	
(first	G 10 0000 0000 0000	
time)	H 00 1501 0121 0002	

StepResult

At end of E 35 1651 1652 1650
 PC 232 F 02 0000 0000 0010
 G 00 0000 0000 0000
 H 00 1502 0121 0001

Results printed out:

1501 35 1651 1652 1650
 1502 36 1001 1002 1003

in PC's 241 through 245, except that the test for the last digit is made against the product F_2G_5 from 0_{11} through 0_1 . This is because the magnitude of F carries the digit which shows the last digit to be typed from a word. Here, this digit appears in 0_1 of F, and thus the printing process is repeated 13 times to print out the entire word in E, except for the last digit. When the 13th digit has been printed, the test for last digit in PC 221 sets K true for skip to PC 225, where A_1-A_6 are set to actuate the Flexowriter "tab" key, following the usual routine in PC's 226-230.

In PC 231, 1 is subtracted from m_3 of H, and 1 is added to m_1 of H. Also P_2^0 of F is tested for normal or error printing. In this case it is 0, and K stays false for count to PC 232 where m_3 of H is tested to see whether the last word has been printed.

After the word in 1501, there is still a 1 in m_3 of H; hence K goes true for skip to PC 236, and the above routine is repeated for printing out (1 502). When (1502) has been printed, K is set false in PC 232 for count to PC 233, a dummy block where K is set true for skip back to PC 6 and the next command shown in m_2 of H (0121).

COMPUTER-IBM CIRCUITRY AND OPERATION

ified as indicated below:

General Information

The Model 102-A is capable of accepting signals read from cards by an IBM Gang Summary Punch Type 523, or equivalent; and the 102-A will provide signals to be punched into cards by another such machine. To accomplish this, the IBM equipment must be mod-

<u>IBM Model</u>	<u>IBM Standard Modification No.</u>
523 Summary Punch	
Input	RPQ37624
Output	RPQ37625
514 Reproducing Punch	
Input	RPQ37626
Output	RPQ37627

In order to secure the desired modification, it is necessary to notify the IBM Corporation that a modification is required. The modification will be made by IBM either before or after delivery of the machine.

Since these modifications are electrical in nature, they may be switched "IN" or "OUT" as desired so that the IBM machines may be used in their normal fashion with other IBM equipment. The circuitry in the computer which accomplishes the read and punch operations is discussed in the following paragraphs.

Preparing Cards in Hopper

The cards to be read are placed in the feed hopper face down, "12" or top edge first. They must be arranged in the order in which the computer is prepared to accept them.

Two cards must be fed into the Summary Punch (manually) before computer operations are started so that the first card will be in position to be read. The Summary Punches used with the Model 102-A will do only one operation, either read or punch; therefore, two machines must be used to have both operations.

The 102-A reads one card at a time. The Summary Punch operates under control of the computer; it does not run continuously unless the computer instructs it to do so.

IBM Control Panel

The IBM control panel is removable and fits into a rack in the front of the machine. The control panel as shown in Figure IV-36 together with the following explanation will indicate the purpose of the wiring hubs.

Punch Brushes: There are 80 outlet hubs for the punch brushes which are connected to the computer through a cable. As shown in Figure IV-36, the punch brush hubs must be jumpered to

the hubs marked with an X for reading octal data and to the hubs marked with a box (□) for reading decimal data.

Octal Data (hubs marked with an X on Figure IV-36): there are 56 inlet hubs for transmitting octal data into the computer and one inlet hub for transmitting a special signal indicating that a card contains octal data.

When the words are assembled in the Buffer Register the first word is read from hubs #3 - #16, the second from #23 - #36, the third from #43 - #56 and the fourth from #63 - #76 of the comparing magnets field. Hub #20 of the comparing magnets field must receive a "12" punch signal from every card which is to be read octally. This is necessary for proper operation of the card reading process.

Decimal Data (hubs marked with a box (□) on Figure IV-36): There are 40 inlet hubs for transmitting decimal data into the computer. They are grouped in four words of ten decimal digits each.

The first decimal digit of each word may be arranged to be a sign digit if desired. A negative sign is indicated as an X punch in a column separate from the data columns. When the words are assembled in the buffer register the first word is read from hubs #4 - #13, the second from #24 - #33, the third from #44 - #53, and the fourth from #64 - #73 of the comparing magnets field.

Reading Procedure in the Computer

The complete reading procedure depends on memory circuitry within the computer, the existence or lack of a "12" punch in a particular column, and on parameters specified by the READ CARD command. The memory circuitry.

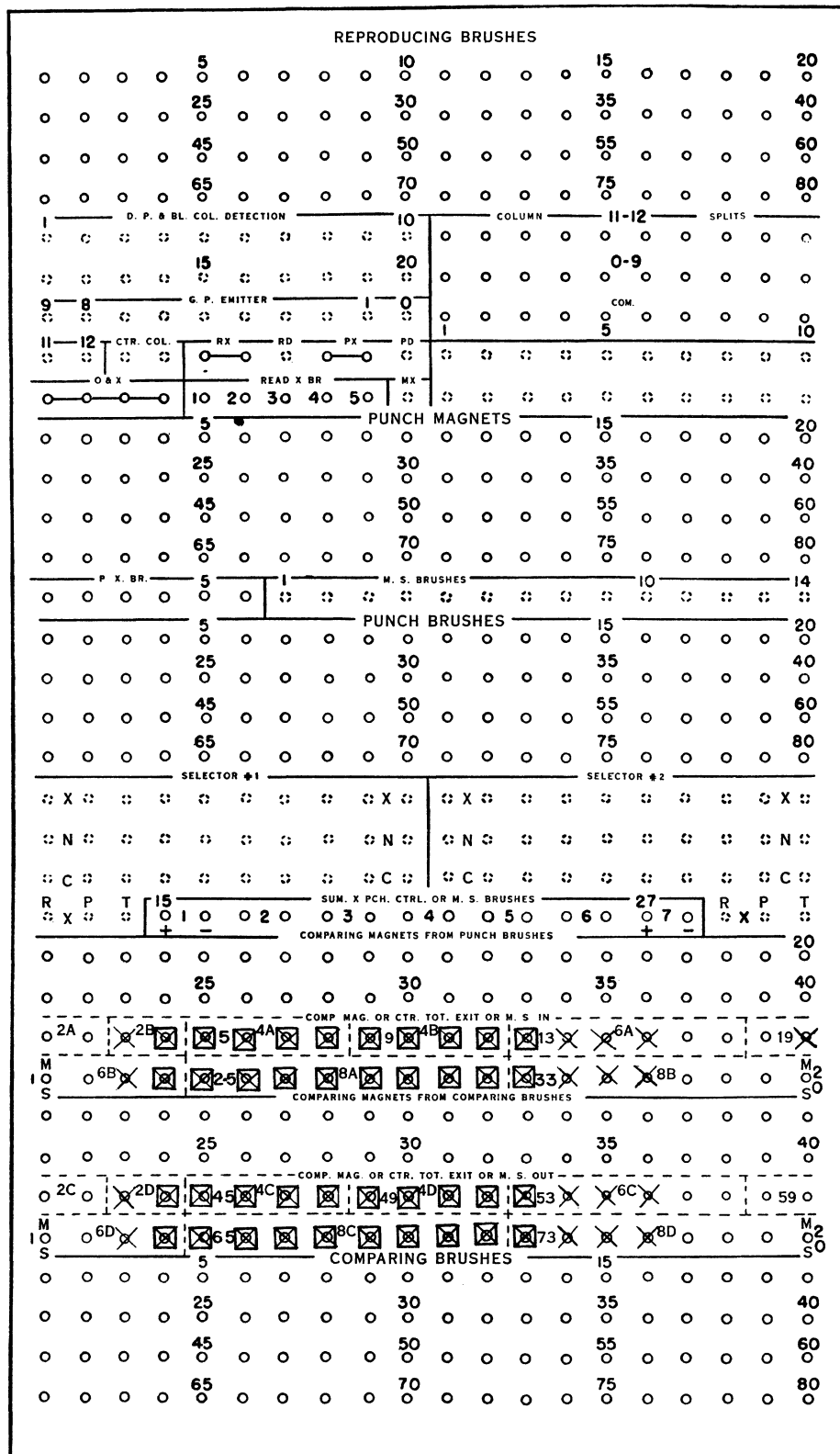


Figure IV-36. IBM Control Panel

involved in this operation requires three separate registers.

The Buffer Register assembles words being read in four adjacent cells, starting at a cell specified in the READ CARD command. Each digit in a word is numbered in a fashion corresponding to the hubs in one row of the IBM control panel.

The Input Register is initially filled by the READ CARD command with a group of digits which are inserted into the corresponding digit positions of the proper words in the Buffer Register when signals are received at the inlet hubs. The Input Register is indexed as each row of the card passes under the reading brushes so that holes in each succeeding row insert the next value into the Buffer Register.

The Addend Register is filled by the READ CARD command with digits which are added to the Input Register while the card moves between holes. The normal configuration will usually be a group of decimal or octal "ones."

The command number for the READ CARD routine, octal digits 06, is placed in the operation portion of the H register, $0_{13}0_{12}$, in Block 10 along with the remainder of the command. As a result, in PC 16 there is a zero in P_1O_{13} of H; and K is set true for a skip to PC 400. In PC's 400 through 404, a binary 1 is added in each block to the command number (000 110), so that the latter becomes 001 010 in 0_{13-12} of H. Now, there is an automatic skip to PC 412, and again the process of adding 1 to the operation portion of H is resumed until there is a 1 in P_1O_{13} (010 000), which occurs in PC 415 and results in setting K true for a skip to PC 455, the start of the card-reading routine.

When PC 455 goes true, a signal is sent through a relay driver to energize the "start read" relay as shown in Figure 26, Vol. II; and this in turn sends a signal to the IBM reader. The cards are so placed in the hopper that the top edge or 12 rows are read first. When the reader is ready to start reading the 12 row of the first card, it sends a return pulse which turns on the reader one-shot in the computer. This one-shot turns K false for a count to PC 456; the one-shot is on for about 3 ms. In PC 456, there is a delay of eight words (3.4 ms) timed by the removal of the markers from G.

PC 457 provides a test for octal or decimal reading. If the cards originally were punched octally, there is a hole at column 20 of the 12 row; and this sets K true for a skip to PC 557. Otherwise, the count is to PC 460.

For illustration, the decimal card-reading routine may be followed, since the octal version is essentially similar. PC 460 is a dummy block to set K true entering PC 461, so that the computer will stick in this latter block until the next reader one-shot signal is initiated, indicating the start of reading the 11 row. Now, the count is to PC 462 where (F) is added to (E). These registers must always contain certain numbers, as specified by the programmer. E contains in each decimal digit the pulse pattern that a hole in a column in each row will represent. F contains the increment by which each digit in E is increased for each row.

Thus E initially contains 01 1101 1101 1101 1101 1101 1101 1101 11 01, ignoring the unused first four bits (all 0); or, for the octal equivalent programmed as (m_1), E contains 0167356 7356736.

Similarly, F must contain (and continue to contain) 00 0001 0001...0001 0001 or for (m_2) 01042104210421 octally.

Thus adding (F) to (E) the first time, E becomes 01 1110 1110...1110 1111. This is the pattern for the 11 row, not normally punched. When (F) is added to (E) the second time, the content of E becomes all zeros, so that the 0 row of the IBM card is represented by a 00 00 in each decimal digit of E. The next time through the routine for the 1 row, (F) will again have been added to (E) and the latter will hold a 0001 in each decimal digit, and so on.

In the same Block 462, the channel selector flip-flops L_5 and L_4 are set to assure selection of the buffer, J, for putaway. PC 463 provides another 8-word delay, and PC 464 determines whether the putaway of the four words from the IBM card will start with buffer cell 2000 or cell 2004.

The computer stays in PC465 for four word times, while data from one row of the card is read into J. A_1 is true the first word time; A_2 true the second word time; A_3 the third; A_4 the fourth. (See explanation in Block Explanation for PC 465 in Volume II) Table IV-12 summarizes this operation and also octal reading (PC 564).

Thus, in reading a decimally-punched card in the 5 row, for example, columns 4-13 are examined and wherever there is a punch, the contents of E, namely 0101, are transferred to J for that decimal digit position, and placed in cell 2000. During the second word time, information is placed in cell 2001, etc.

PC 466 provides a test for the last row.

First Word (A_1 true)	Second Word (A_2 true)
IBM 3.....16 (octal)	23.....36 (octal)
Col- 4.....13 umns (decimal)	24.....33 (decimal)
Third Word (A_3 true)	Fourth Word (A_4 true)
IBM 43.....56 (octal)	63.....76 (octal)
Col- 44.....53 umns	64.....53

TABLE IV-12. Reading a Row of an IBM Card

In decimal card reading, when E contains 1001 in the last four bits, the 9 row has been read, and the contents of the card have been transferred to the buffer. Then, there is a count to PC 467 where K is set true for a skip back to PC 6 for the next command. If E does not contain binary 9's, there is a skip back to PC 461 to read the next row, and this continues until the entire card has been read.

Reading of octally-punched cards, Blocks 557 through 566, follows the same pattern except that octal digits, each with three bits, are used: 14 columns are read at a time and the original contents of E and F are correspondingly different. The test in PC 565 is for octal 7's in 0_0 of E, to indicate the end of reading the card. When the cards are punched octally, the 8 and 9 rows are not used.

Note that reading one card, either decimally or octally, will fill our buffer cells. Thus, two IBM cards can be read into the buffer before this J line must

be unloaded. Hence, in preparing a program to read these cards, the programmer must unload the buffer (bo, PC417, 4361447) into the main memory after every two read-1-card (rc) commands. Because the routine is repetitive, a large number of IBM cards can be read into the memory with the use of only five commands by the programmer.

Punching Procedure in the Computer

The complete punching procedure depends on fully automatic apparatus within the computer, whether the command is PUNCH OCTAL or PUNCH DECIMAL, and on parameters specified by the commands. Three registers in the computer are involved in this automatic process.

The Buffer Register must contain the words to be punched in four adjacent cells starting with the cell specified by the punch command. Each digit in a word is numbered in a fashion corresponding to the hubs in one row of the panel.

The Comparison Register is initially filled by the punch commands with digits which are compared to the digit positions of the proper words in the Buffer Register. When a digit in the Buffer Register compares to a digit in the Comparison Register, a signal is transmitted to the hub on the control panel corresponding to the digit position in the Buffer Register. The Comparison Register is indexed as each row of the card passes under the punch dies so that each digit is punched in the proper row of the card.

The Addend Register is filled by the punch commands with digits added to the

Comparison Register while the card moves between holes. The normal configuration will be a group of decimal or octal "ones".

The routines for both octal and decimal punching can be understood by studying the Block Explanations for PC 412, 520-537, 500-517 (see Volume II). Note that either PC 500 or PC 520 will turn on the relay driver, energizing the "start punch" relay in the computer which sends a signal to the Summary Punch.

In the punching routines, the punch one-shot in the computer is triggered by the Summary Punch at the start of each new row. The delays utilized in PC 525 (eight-word delay), PC 526 (wait for proper word from buffer), PC 527 (four-word delay) and similar blocks in the decimal punch routine are provided primarily to maintain identical timing for each row of the card.

Thyratrons are set up in the computer to provide the actual impulses for punching each digit space on a card. Since punching a card octally is accomplished in four word-times, with a group of 14 columns available for punching during each word-time, 56 thyratrons are required for this purpose. Furthermore, the "12" row must be punched in column 20 to indicate that the card has been punched octally, and this calls for a 57th thyratron. For punching cards decimally, four groups of 10 columns each, corresponding to 10 decimal digits per word (as before in card reading), are punched. Thus only 40 of the 57 available thyratrons are required.

The termination of the automatic process for punching cards depends on the contents of the Comparison Register.

After each row of the card is punched, before the Addend Register is added on to the Comparison Register, the least significant digits of the Comparison Register are examined by the machine. If the command was PUNCH DECIMAL, the automatic punching process will terminate when the least significant decimal digit is 9: i.e., the binary configuration is 1001. If the command was PUNCH OCTAL, the automatic punching process will terminate when the two least significant octal digits are 7: i.e., the binary configuration is 111 111. A general comment is that operations take place much more rapidly in the 102-A Computer than in the IBM Summary Punches, and hence numerous program counter blocks are utilized in both the reading and punching of cards to wait for IBM signals.

COMPUTER - TAPE UNIT CIRCUITRY AND OPERATION

General Information

The Model 126 Magnetic Tape Handling Unit provides additional storage space on its tape - equivalent to 100 times the capacity of the memory drum of the 102-A - for applications which require exceptionally long programs, many programs, or the storage of large quantities of tabular data. As many as seven tape units may be connected to the 102-A simultaneously. The following paragraphs describe the tape routines in the computer. For more detailed information on the operation of components in the tape unit the reader is referred to the "Model 126 Magnetic Tape Handling Unit - Operation and Maintenance Manual".

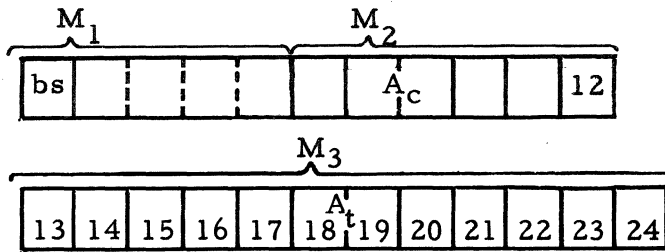
The Tape Unit has three modes of op-

eration: Searching, reading, and writing. In searching the unit operates at high speed (90 inches per second) and travels forward until the address is equal to the desired address; or, if the first address read is greater than the desired address, it reverses to find the block. In reading and writing the tape moves forward at low speed (15 inches per second) and traverses one block at a time.

There are 7050 blocks on a 1200 ft tape, with octal addresses in the range from 00000 to 17777. There are 17 words per block, including one address word and 16 information words. The Model 102-A Computer reads or writes only 8 words (the Buffer capacity) at a time from a block: either the odd group (words 1, 3, 5, ... 15) or the even group (words 2, 4, 6, ... 16).

Information is read from, or written on, the tape in a series-parallel arrangement. Four information channels are recorded in duplicate on eight tracks. Two tracks also are available for the tape clocks. In each tape clock period, four bits of information are read from or written on the four parallel information channels, so that data are transferred to and from the computer in groups of four bits. TABLE IV-13 shows how the bits are arranged in the address word on a tape block. Here, T_1 through T_4 are the four information channels on the tape, D_2 through D_6 are digit propositions representing five of the 11 digit times in a word, and 12' through 24 are relays for the 12-bit register holding the block address.

The same address, as it would appear in a 102-A word, is shown as follows:



The circuits required for the information transfer between the computer and tape unit(s) are shown in Figure 26, Vol. II.

	D ₂	D ₃	D ₄	D ₅	Forward	
	LSD		MSD			
T ₁	24	20	16	12		
T ₂	23	19	15	N	11	U
				0		S
T ₃	22	18	14	T	10	E
T ₄	21	17	13		9	D
	D ₆	D ₅	D ₄	D ₃	Reverse	

TABLE IV-13. Arrangement of Bits in Tape Address Word

Block Search Routine

In the Block Search routine, the signal to search sent by the computer is generated by PC 406. When this program counter number goes true, its 125v signal is connected to the tape unit. This proposition is referred to as S_c . It causes Relay R_t in the tape unit to close. The proper tape unit is selected by the information contained in computer flip-flops A_9 - A_{11} . These flip-flops operate relay driver UTC2-A3 which causes proposition T_n to go true in the correct tape unit.

In PC 403, the operation portion of H becomes 01 000; and K is set false for a skip to PC 405 where the number of the tape unit to be searched is transferred from 0_6 of H to flip-flops A_9

through A_{11} . The tape block number, as described previously, is transferred from the last five octal digits of H (0_0 - 0_4) to flip-flops A_7 - A_8 (MSBD), L_1 , L_4 , and A_1 - A_6 (LSBD). Now, the count is to PC 406. The computer sticks here until a relay signal from the tape unit is received to set K false for a count to PC 407. Meanwhile, the computer has sent proposition S_c , the signal to search, to the tape unit which closes the appropriate relays and transfers the 13 bits held in the computer's A_1 - A_8 and L_1 - L_5 flip-flops to the address relays 12-24 in the tape unit. Now R_{sn} in the tape unit signals the computer that the set-up for block search is complete, and this releases the program counter in the computer to count to PC 407 while the tape unit goes ahead independently with its block search.

In PC 407, the control number which was stored in the G register is transferred to H, and the program counter skips back to PC 6 for the next command.

Read Tape Routine

In the Read Tape routine, the command is detected in the usual way, and this results in a skip from PC 401 to PC 601. Here, information from m_2 and m_3 of H is transferred to the appropriate flip-flops: 0_6 contains T, and the three bits of this octal digit go to A_9 - A_{11} . The mode information K from 0_4 goes to A_7 and A_8 , while m_3 of H, containing N (the last digit of the address of the first buffer cell to be read into), is transferred to G. In PC 605, the counter sticks until a tape clock pulse is received, along with the four parallel bits of information from the tape which are read into L_1 - L_4 . During PC's 606

through 620, information goes from the tape to L_1 - L_4 , from these flip-flops into the A_1 - A_4 flip-flops, and thence into the E register (see Figure IV-37). This order continues until a complete word has been transferred. In PC 621, the word goes from E into the correct cell of the Buffer register, J. As shown in the Block Explanation Forms for PC's 622 and 623, this process is repeated until the eight words (odd or even) of the tape block have been read and the buffer is loaded. Of course, in mode 0, requiring only the reading of the tape block address, the read routine is completed after the reading of the single address word.

Write Tape Routine

The Write Tape routine is similar to the read routine (see Block Explanation

Forms PC 644 to 673 in Vol. II).

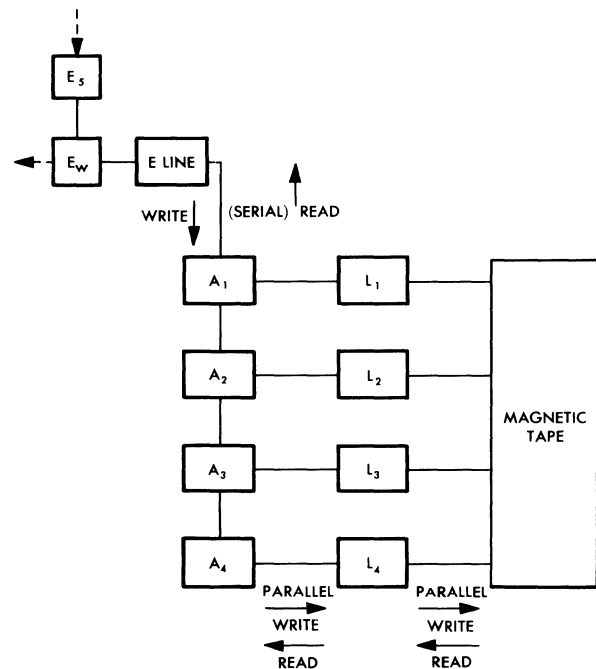


Figure IV-37. Block Diagram of Computer Tape Information Transfer

Operation

INTRODUCTION

This section describes in detail the complete operating procedure for the Model 102-A computing system. This includes the functions of all controls necessary to operate the computer and a brief introduction to programming techniques. This latter will serve to acquaint the operator with some of the common short cuts which are helpful and necessary for efficient use of the computer. An understanding of the following text will enable the reader to operate the complete 102-A system, including all the possible input-output devices available with the system.

CONTROLS (Operator's)

Computer Control Panel

The computer control panel is mounted on the Operator's Console. The six push buttons, seven toggle switches, and seven indicator lights are arranged as shown in Figure V-1. They perform the following functions:

PUSHBUTTONS

POWER ON -- applies power to the computer and all magnetic tape units connected to the computer which are switched to AUTO.

POWER OFF -- turns off all power in the computer and in tape units connected to the computer which are switched to AUTO. The **MEMORY SAFE** switch operates as an override and must be in the Down position before this button will function.

BASE 10 -- prepares the computer to accept decimal numbers.

BASE 8 -- prepares the computer to accept octal numbers and commands.

COMPUTE -- starts actual computation.

CLEAR -- resets computer to idle state.

NOTE

This button may not be used as an operator halt; computation may not be restarted where it was stopped. If pressed during the middle of a computation, it will immediately halt computation, possibly during the execution of a command, and place the computer in the IDLE condition ready to accept octal input. It does not clear data out of any portion of the main memory, buffer register, control register, or input register.

SWITCHES

MEMORY SAFE -- operates to protect information stored in the memory against an accidental recording during the power on or off sequences. In the Up position, the **POWER ON/OFF** push-buttons are made inoperable as far as the computer is concerned. In the Down position, it automatically activates the **CLEAR** pushbutton and allows the d-c voltages to be applied or turned off.

FILL -- permits information typed on the Flexowriter to be entered into the computer when the switch is in the Up

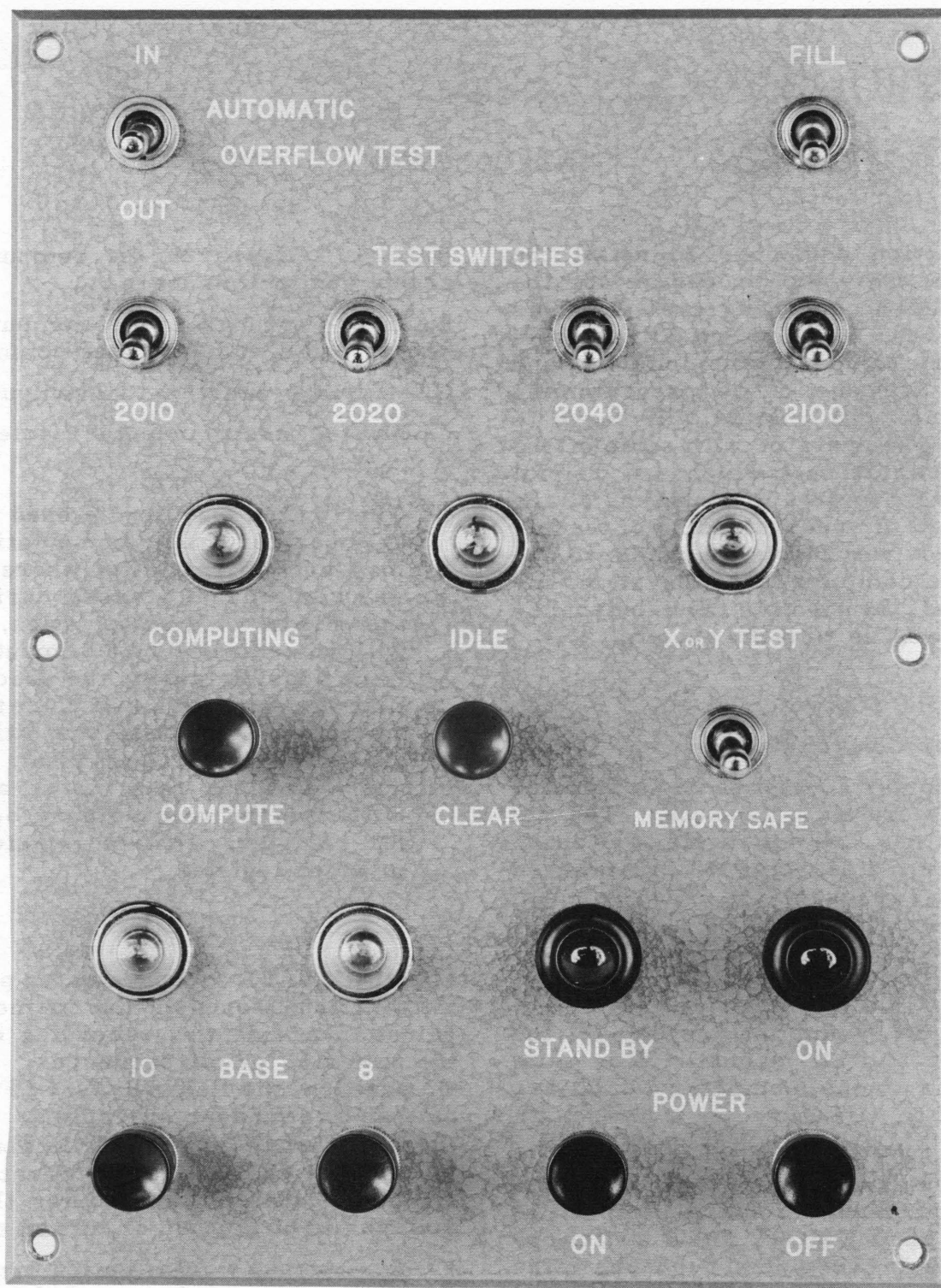


Figure V-1. Computer Control Panel

position. In the Down position, the information typed on the Flexowriter will not enter the computer; however, the computer remains connected to the Flexowriter and may direct it to type output results.

AUTOMATIC OVERFLOW TEST -- controls the overflow checking feature. If an arithmetic operation gives a result equal to or greater than one in magnitude, and the automatic overflow switch is in the IN position, the contents of the G register will be printed out and the computer will halt, unless the next command is **TEST FOR OVERFLOW (to)** or **SHIFT LOGICAL (sl)**. In the OUT position, the overflow checking feature is suppressed and the overflow is not included in the result.

TEST SWITCHES -- labeled 2010, 2020, 2040, and 2100 are used in conjunction with the **TEST SWITCH** command and represent addresses in the memory containing permanently recorded coded information. Their particular functions are described in detail in SECTION III.

INDICATOR LIGHTS

STAND BY -- operates in conjunction with the **POWER ON** push button to inform the operator that the power-on sequence is in process and that the computer is warming up.

ON -- informs the operator that the power-on sequence is complete and that the computer is ready for operation.

BASE 10 -- operates in conjunction with the corresponding pushbutton and the "d" key on the Flexowriter, informing the operator that the computer is prepared to accept decimal numbers.

BASE 8 -- operates in conjunction with the corresponding pushbutton and the "o" key on the Flexowriter, informing the operator that the computer is pre-

pared to accept octally coded information.

IDLE -- indicates that the computer is ready for operation but is not computing.

COMPUTING -- indicates that computation is in process.

X or Y TEST -- indicates that some switch on the computer test panel is thrown and must be reset before the computer will operate properly.

Flexowriter

The Flexowriter, mounted on the Operator's Console, is the basic input-output device for the computer (see Figure V-2). The following keys on the Flexowriter are significant to the computer when the fill switch on the Computer Control Panel is in the Up position and the IDLE indicator light is on:

CONTROL KEYS

d -- prepares the computer to accept decimal numbers and clears the E register. Once the "d" key has been struck, any significant Flexowriter key will cause four bits per character to enter the E register (see TABLE V-1).

o -- prepares the computer to accept octal numbers or commands and clears the E register. Once the "o" key has been struck, any significant Flexowriter key will cause three bits per character to enter the E register (see TABLE V-2). It is noteworthy, however, that the computer is normally prepared for octal input.

-(hyphen) -- transfers information from the E (input) register to the H (control) register and clears the E register. This

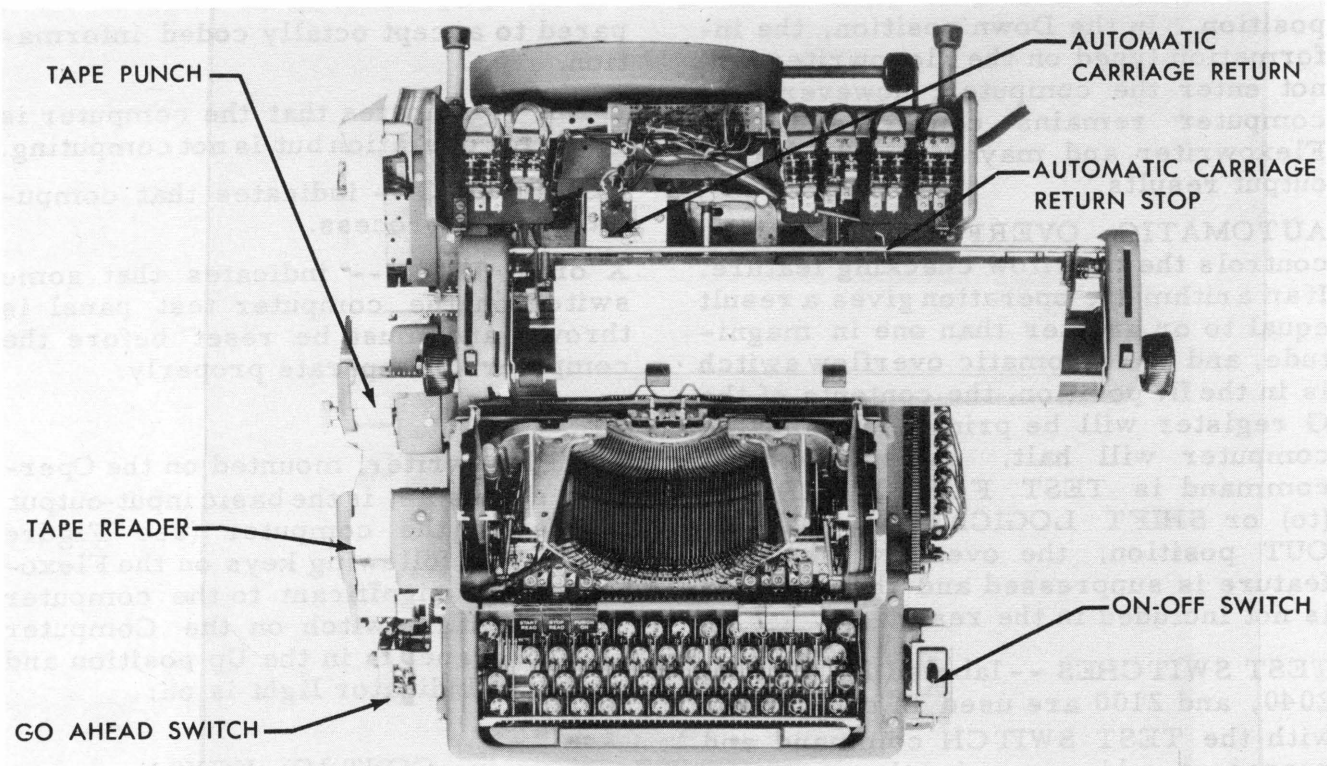


Figure V-2. Flexowriter

permits placing a monitoring address in H at any time.

"Tab" -- transfers the contents of the E register to the memory cell named by the address in the m_3 portion of the H register. Striking the "Tab" key also clears the E register, and augments the current address in m_3 of H by one. The fact that m_3 of H is automatically increased by one when the Tab key is struck permits information to be filled into sequentially numbered memory cells when only the initial address is placed in H.

f -- fills the equivalent of four octal or four decimal zeros, depending on the mode of input, in the E register. Striking the "f" key enters one zero and signals the computer to circu-

late the contents of E left three additional digit positions.

NOTE

Care must be exercised when using the "f" key. It will fill four octal or decimal zeros only if the three most significant octal or decimal digits of the word are zero. Basically, this is because striking the "f" key inserts a zero, and then circularly shifts the register four octal or decimal places to the left. If a filling error is made, the Input register cannot generally be cleared by repeatedly striking the "f" key.

s -- controls the start of machine computation. When the "s" key is struck, computer operation will be initiated according to the address in m_2 of H.

NUMBER KEYS

These keys are used to fill either decimal or octal numerical data. When the BASE 10 indicator light is lit, these keys fill decimal numbers into the machine (see TABLE V-1).

KEY	BINARY CONFIGURATION
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
SPACE BAR	1100
PERIOD	1111
+	0000
- (minus key)	0010
f (fills four decimal zeros)	

TABLE V-1. Decimal to Binary Conversion Code

When the BASE 8 indicator light is lit, the number keys fill octal numbers into the machine (see TABLE V-2).

KEY	BINARY CONFIGURATION
0	000
1 (L key)	001
2	010
3	011
4	100
5	101
6	110
7	111
f (fills four octal zeros)	

TABLE V-2. Octal to Binary Conversion Code.

Modifications to the Flexowriter effect only the keys mentioned previously, all other keys, switches, and control buttons operate in the conventional manner as described in the "Flexowriter Operating Instructions Manual". The control and number keys are the only Flexowriter characters that are significant to the computer.

For convenience in input and output the Flexowriter is equipped with a special tab stop on the rear of the carriage. This stop can be adjusted manually to cause the carriage to return to any desired point across the line.

Attached to the side of the Flexowriter unit are the paper tape "punch" and "read" units. The controls for these units are mounted as shown in Figure V-2. Preparation of the paper tape is identical to the manual filling process. Once the PUNCH ON control is depressed, striking any Flexowriter key will automatically punch a distinct configuration of holes in the paper tape as shown in Figure V-3.

A READ TAPE control on the Flexowriter permits punched paper tape to be fed into the Flexowriter and communicate with the 102-A. Paper tape feeding may be halted automatically by a STOP CODE punch on the tape. A punched paper tape, then, can fill the computer, start computation, and halt itself from further feeding.

The principal feature of using punched paper tape is that the tape can be prepared on a separate Flexowriter unit, and it can be proofread and corrected prior to its being fed into the computer. Hence, new tapes can be prepared and checked while computer operation is in progress. Also, paper tape makes a permanent and accurate copy of a routine available for future use.

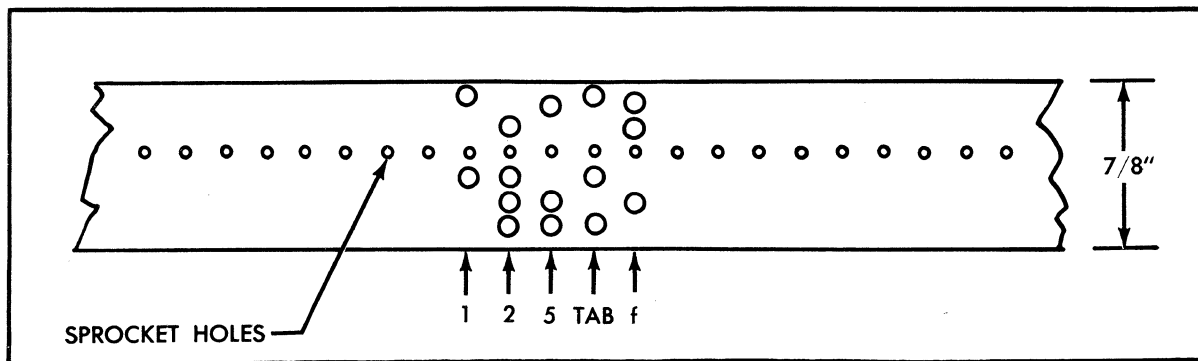


Figure V-3. Flexowriter Character Punched on Paper Tape

Test Panel

The Test Panel is mounted on the front side of the computer. It contains various switches, meters, indicator lights, and test jacks which enable the operator to check d-c voltages, skip the computer from rest to any program counter number or from any program counter to rest, turn ON or OFF the d-c voltages, and obtain waveforms from within the computer (see Figure V-4). The following text describes the function of each switch, meter, light, and test jack:

SWITCHES

DC ON/OFF -- applies or disconnects the d-c voltages. This switch will not operate unless the **MEMORY SAFE** switch is in the Down position.

MEMORY SAFE -- operates to protect information stored in the memory against an accidental recording during the power on or off sequences. In the Down position, it performs approximately the same function as the **CLEAR** button on the computer control panel; that is, it holds the clear proposition true. In addition, it cuts off all power to the record propositions so that no stray transients can develop in the record circuitry and thus erase or alter information in the main memory when the DC switch is operated. In the Up position, this switch prevents the DC

OFF/ON switch from operating.

X ON/OFF -- controls the X test function. When the X switch is ON, the computer will go through the normal sequence of operations, according to the program command applying at the time, until it reaches the program block designated by the N_1 to N_9 switch register for the X test function; at which time, regardless of program, it will skip back to rest.

Y ON/OFF -- controls the Y test function. When the Y switch is turned ON, and a block has been set up on the N_1 to N_9 switch register for the Y test function, the computer will skip directly to this block and then continue its normal sequence, according to the program command in operation.

K/K' -- is used in conjunction with the Y test function to set the decision (K) flip-flop in the correct configuration to enter the block designated by the N switches.

N_1 to N_9 -- switch registers consist of two banks of toggle switches, with nine switches in each bank, labeled N_1 , N_2 , N_3 , ..., N_9 . The top bank is used in conjunction with the X test function, and the lower bank is used with the Y test function. It is possible to set up the octal representation of any program

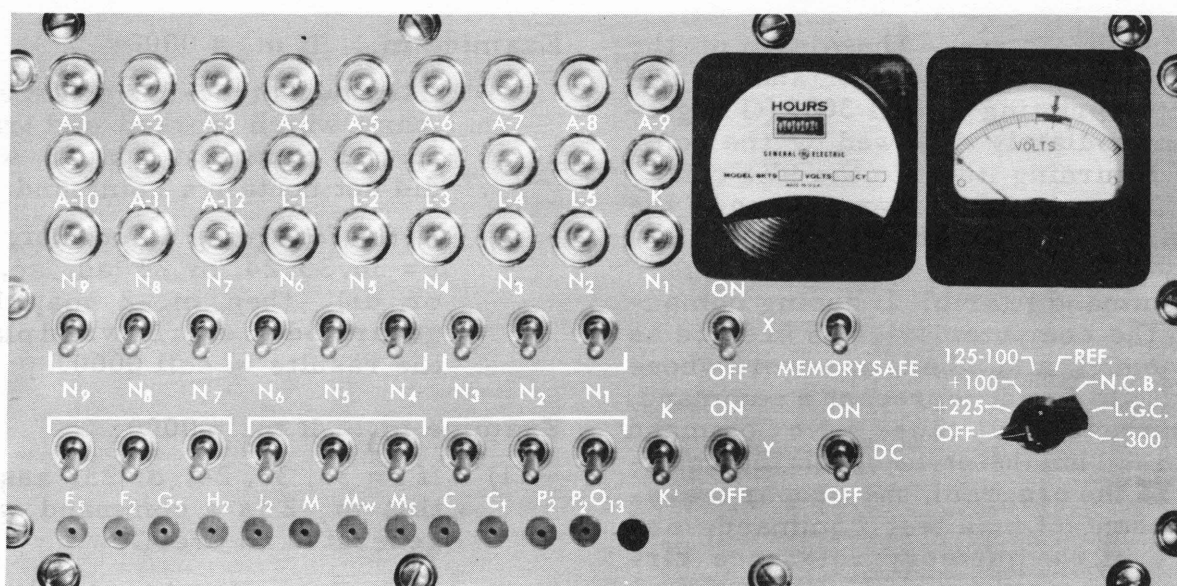


Figure V-4. Computer Test Panel

block number on either bank by operating the switches: UP = 1, DOWN = 0.

METERS

VOLTMETER AND EIGHT-POSITION SELECTOR SWITCH -- are used to check the d-c voltages in the computer. The voltmeter is calibrated to indicate whether or not the particular voltage is within the specified tolerance.

TIME METER -- indicates the number of hours the filaments of the tubes in the computer have been energized.

INDICATOR LIGHTS

The top two rows of indicator lights inform the operator of the states of flip-flops A_1 through A_{12} , used for miscellaneous storage and control functions; the channel selector flip-flops, L_1 through L_5 ; and the decision and carry flip-flop, K. The third row of indicator lights refer to operation of the program counter flip-flops N_1 through N_9 .

TEST JACKS

The row of test jacks permit examination of the waveforms at various internal test points. E, F, G, and H can be connected to an oscilloscope to test the contents of these one-word registers; J jack shows the contents of the eight-word buffer; M and M_w are the main memory and the word channel, respectively (M_s , this test point is no longer used); C_s , the computer clock; C_t , the magnetic tape unit clock; P_2^0 , a synchronizing marker that is the digit trigger or marker for Z axis to mark octal digits across a word; and $P_{2^0}^{13}$, a pulse that is the word trigger. Each jack is connected through a blocking capacitor and, therefore, some waveform distortion takes place.

CONTROLS (Automatic)

Alarm Checks

The Model 102-A Computer is provided with two automatic alarm checks as indications of programming errors and as some protection against improper

machine operation. The Alarm on the 102-A consists of the spontaneous unexpected printing of cell 3000 (G register) immediately followed by the computer returning itself to the idle condition. The special interpretations of the alarm are indicated below.

No Command Alarm: If during computation the computer tries to execute as a command, a number, or a word whose sign digits do not represent a command, the computer will cause a No Command Alarm. This helps to check the continuity of the program, the proper transfer of control in a test command, and failure of the memory reference circuitry of the computer.

Overflow Alarm: If an ADD, SUBTRACT, or DIVIDE command causes an overflow as described in their respective sections, and the command is not immediately followed by a TEST FOR OVERFLOW MARKER or a SHIFT LOGICALLY command, and the Automatic Overflow Test switch on the Computer Control Panel is in the IN position, the computer will cause an Overflow Alarm. It is not necessary that the TEST FOR OVERFLOW or SHIFT LOGICALLY commands operate on the overflow result produced by the command being performed at the time. This helps to check the scaling of the problem.

The interpretation of cell 3000 when printed as an alarm is as follows:

I m_1 m_2 m_3

The use of I, m_1 , m_2 , or m_3 in this discussion refers to the correspondingly designated octal digits when printed out as illustrated above. The alarm occurs because the proper contents were not found in cell m_2-1 . The diagnosis of cell 3000 provides an indication of the contents of m_2-1 or the command which led to it.

Examine m_3 . If $m_3 = 0000$:

- 1) Assume that m_1-1 is a test command which worked and transferred control to m_2-1 which did not contain a command.
- 2) If the foregoing is unlikely and $I = 36, 35, 24$, or 23 (ad, su, dd, or dr), then m_2-2 may have generated an overflow and placed the results in cell 0000.

Examine m_3 . If $m_3 \neq 0000$:

- 1) If $I = 36, 35, 24$, or 23 , assume that m_2-2 is a command which generated an overflow.
- 2) If the foregoing is unlikely, assume m_2-1 is not a command.
- 3) If $I = 36, 35, 24$, or 23 , m_2-1 is not a command.

Experience in operating the computer will provide the user with a certain proficiency in diagnosing program alarms.

FILLING THE COMPUTER

FLEXOWRITER

The Flexowriter is the primary input device associated with the computer. The following paragraphs apply to both automatic (paper tape) and manual filling, since the paper tape must be provided with all the symbols which the operator would use to enter the data. Either method will produce hard copy and, at the option of the operator, punched paper tape. Thus, in the case of automatic filling, a duplicate tape could be punched.

Manual Filling

There are two filling modes, octal and decimal, and it is important that the proper mode be selected prior to filling. The two registers (E and H) which are used in the filling operation are both

one-word delay lines. The input register (E) accepts and assembles individual digits directly from the Flexowriter. Each new digit, as it is entered, causes all previous digits in the register to move one digit to the left and the new digit itself is entered in the rightmost position. The control register (H) controls the dispatching of the assembled word to its proper destination cell and contains the address of the next command in order. For example, consider filling the decimal number 3.1416:

- a) First the input register is cleared and the "d" key is struck on the Flexowriter.

0 0 0 0 0 0 0 0 0 0

- b) Strike the 3 key

0 0 0 0 0 0 0 0 0 3

- c) Strike the period (.) key

0 0 0 0 0 0 0 0 0 3 .

- d) Strike the 1 key

0 0 0 0 0 0 0 0 3 . 1

- e) Strike the 4 key

0 0 0 0 0 0 3 . 1 4

- f) Strike the 1 key

0 0 0 0 0 3 . 1 4 1

- g) Strike the 6 key

0 0 0 0 3 . 1 4 1 6

The contents of the input register are transferred to the control register by hitting the hyphen (-) key on the Flexowriter. The information is inserted in such a way that the address of the next command is held in the m_2 position of H; and the destination of the next word to be dispatched is held in the m_3 position. For example, if the control register held

0 0 0 0 0 0 1 6 4 2 0 7 3 5

The next piece of data dispatched would go to cell 735, or if computation were

started the computer would take its first command from cell 1642.

To continue the filling process, select the octal number system in order to fill the control register with the address of first cell to be filled (this may be verified by observing the indicator light marked BASE 8).

NOTE

CLEAR button on the Computer Control Panel will set computer to fill in octal.

If computer has been computing and has been stopped either by a HALT or FILL command, it is already set to fill in octal.

If decimal data have just been filled, it is then necessary to press the "o" key or the BASE 8 pushbutton on the Computer Control Panel.

Next, select the address of the first cell to be filled. If the computer has been computing and stops for a HALT or FILL command, the m_3 position of either of these commands will be the cell where input will start if the control register is not modified. To specify a particular first address to the control register, the desired octal address, followed by a hyphen, is typed on the Flexowriter.

At this point the appropriate number system should be selected, either octal or decimal depending on the data to be entered. Once a number system is selected, it is preserved until changed.

Enter data as follows:

Commands - octal system

- a) Enter the two digits identifying the command.
- b) Enter the three addresses. Where one of the addresses is not significant to the command, it

may be filled with any desired value. Thus, four zeros, where needed, may be filled with any desired value; or 2100 or 3000 may be entered for minimum access coding.

Octal Numbers

- a) Enter octal sign. If the sign is positive without overflow, it need not be entered. If any of the other signs are intended they must be entered, as shown below:

DESIRED SIGN	FILLING OPERATION
positive, no overflow	- enter nothing or a zero
negative, no overflow	- enter a 2
positive, overflow	- enter a 1
negative, overflow	- enter a 3

- b) Enter octal magnitude. If the sign was positive without overflow, zeros on the left which would have been needed to fill out the 12 octal digits of the word may be ignored; however, if any other was entered, all 12 digits of magnitude must be entered.

Decimal Numbers

- a) Enter decimal sign. If sign is positive without overflow, it need not be entered. If any of the other signs are intended, the sign digit must be entered as shown below:

DESIRED SIGN	FILLING OPERATION
positive, no overflow	- enter nothing, a zero, or a +
negative, no overflow	- enter a 2 or a (-) minus
positive, overflow	- enter a 1
negative, overflow	- enter a 3

- b) Enter decimal magnitude. If the

sign was positive without overflow, zeros on the left which would have been needed to fill out the 9 decimal digits of magnitude may be ignored. If any other sign was entered, all 9 digits of magnitude must be entered.

When the word has been typed in the input register and the control register has been filled, the contents of the input register can be transferred to the previously-selected destination cell by striking the Tab key. This moves the typewriter carriage, dispatches the word to its destination, indexes the control register to prepare to dispatch the next word to the next consecutively numbered memory cell, and clears the input register.

In the case of an error before the Tab key is struck, it can be corrected by entering the entire correct command or number again, including both sign digits and all leftmost zeros, and then tabing. The following procedure is an alternative method for correcting an error before the Tab key is struck: Strike the "d" or "o" key, depending on the filling mode in use. This clears the input register, and the desired correct information may be retyped. This eliminates the need to fill the entire word including the sign digits if normal filling would not so require.

If an error is detected after tabing, it can be corrected by selecting the octal mode (if not already selected), setting up the control register to the address of the cell where the incorrect data were entered, and re-entering the corrected data.

In the case of a gap in consecutiveness of the data, the first register of the next group of data is selected by setting up the control register.

Automatic Filling

Using a prepunched paper tape to fill the computer may be accomplished by starting the paper tape reader manually or by having the computer start the paper tape reader automatically with the FILL command. When the paper tape is to be used with no manual intervention, after being started either manually or automatically, it must contain certain special characters which control it and the computer.

Specifically, these characters are:

s - Start Computation

Stop Code - Stops the Paper Tape Reader

MODEL 126 MAGNETIC TAPE UNIT (readout)

The Model 102-A drum provides sufficient internal storage for the execution of most problems. However, the drum is limited; and for those applications which require exceptionally long programs, or many programs, or large quantities of tabular data, an automatic data storage medium external to the computer is necessary. The Model 126 Magnetic Tape Handling Unit operates with the Model 102-A for this purpose. As mentioned previously, the Model 102-A is capable of having seven of these tape handling units connected to it at one time.

Data is filled into the computer from a tape unit or units according to the following procedure:

Set selector switch on tape unit(s) control panel(s) to AUTO. Program a BLOCK SEARCH command into the computer (see SECTION III). The Model 126 Magnetic Tape Handling Unit will search for a specific block on the tape independently of the computer.

Program a READ TAPE command into the computer (see SECTION III). Only one block is read with one READ TAPE command. If more than one block on the tape is required, the reading of each block must be preceded by a repeat of the command. In addition, only eight of the sixteen information words in a block may be read with one READ TAPE command.

IBM EQUIPMENT (readout)

The Model 102-A will accept signals read from cards by an IBM Gang Summary Punch, Type 523 or equivalent. The Summary Punch is prepared for operation by connecting the Summary Punch cable to the receptacle on the computer, inserting the control panel in its rack in front of the machine, throwing control switches on the panel to computer operation position (these switches in the modified circuits), and turning the Main Line switch ON.

- a) The cards to be read are placed in the feed hopper face down, "12" edge first.
- b) The START button on the IBM machine is depressed for two card cycles to place first card under reading brushes.
- c) Using the Flexowriter, fill the program containing the READ CARD command into the computer.
- d) Clear the Buffer register by entering "b1 2100 2100 2100".
- e) Start computation.

Example:

Assume the READ CARD command is "rc" 0201 0202 2003; $(m_1) = (201) = 15\ 6735\ 6735\ 6736$ (octal). The binary representation would be

00	1101	1101	1101	1101	1101	1101
----	------	------	------	------	------	------

1101	1101	1101	1110
------	------	------	------

which may be rather loosely considered as being the decimal number

13	13	13	13	13	13	13	13	13	14
----	----	----	----	----	----	----	----	----	----

$(m_2) = (202) = 01\ 0421\ 0421\ 0421$ (octal).
The binary representation would be

00	0001	0001	0001	0001	0001	0001
----	------	------	------	------	------	------

0001	0001	0001	0001
------	------	------	------

which is the decimal number

1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

$(m_3) = 2003$ which indicates that cell 2003 will be the first cell filled and the other three in order will be 2004, 2005, and 2006. This section of the Buffer register is cleared by the usual method.

The READ CARD command is used with both octally and decimally coded cards. (m_1) and (m_2) are specified in each case by the programmer, but the computer operates as if a card is decimal unless a "12" punch exists in a particular column. The octal mode inserts each octal digit into three binary digits and the decimal mode inserts each decimal digit into four binary digits. In this example, there is no wire connected to hub No. 20 so that the data will be assembled in the decimal fashion.

The progressive contents of the Buffer Register and the Input Register as the card moves across the reading brushes and each row of holes in read, is outlined in TABLE V-3.

The termination of the automatic process depends on the contents of the Input Register. After each row of the card

is scanned, before the Addend Register is added to the Input Register, the least significant digits of the Input Register are examined by the machine. If the reading mode is decimal, the automatic reading process will terminate when the least significant decimal digit is 9: i. e., the binary configuration is 1001. If the reading mode is octal, the automatic reading process will terminate when the two least significant octal digits are both 7: i. e., the binary configurations are 111111. If a card is multiple-punched in a column being read, the largest digit will be entered into the Buffer Register. Actually, each successive digit enters its configuration with the last configuration remaining in the end. Note that an X punch cannot be used for a negative sign in the octal mode.

READ OUT FROM THE COMPUTER FLEXOWRITER

Computation results, previously stored permanent data, program monitoring information, and computer alarms are printed out on the Flexowriter. Except for the alarms, all printing is under control of a program. Printout may be in the form of hard copy or hard copy with punched tape.

Data may be encoded for printing in any of the four normal forms previously mentioned: octal numbers, octal commands, decimal numbers, or Flexowriter code. The printing automatically interprets the binary coding of the data and types out the proper character indicated by the code. There is a single PRINT command used for all printing operations (see SECTION III). This command operates as follows:

- a) The total number of words to be typed out with the one PRINT command is m_3 taken as an octal integer; m_3 must be greater than

zero.

- b) The first word to be typed is (m_1). If more than one word is typed, the following words are ($m_1 + 1$), ($m_1 + 2$), ..., ($m_1 + m_3 - 1$). All words are typed in the same mode as the first word.
- c) The printing mode (octal, decimal or alphabetic) is controlled by the sign digits of (m_2):

<u>m_2 sign</u>		<u>Mode of Printing</u>
00	octal	Type the octal sign digits and the magnitude as an octal number.
01	decimal	Type the sign digit as +, -, p, n, period (.), or space according to TABLE V-1 and type the magnitude as a decimal number.

Of the sixteen possible binary configurations, there are four which may not be filled into the computer in the decimal filling process. However, they might appear when printing out, as the result of an error or as extraneous digits. They are interpreted by the Flexowriter as follows:

<u>Character</u>	<u>Flexowriter Interpretation</u>
1010	Code delete (Punch all zeros on tape if tape punch is on).
1011	Ignore
1101	Back Space
1110	Ignore

<u>m_2 sign</u>		<u>Mode of Printing</u>
02	octal with address	Type the octal address of the cell being printed followed by a space, and type its con-

tents as in 00 previously.

- 03 decimal address Type the octal address of the cell with being printed followed by a space, and type its contents as in 01 previously. After printing each word, these four modes of typing automatically tab the Flexowriter carriage to the next tab stop preset by the operator.

m_2 sign
10

Mode of Printing
The mode of alphabetic printing allows the program to activate any key on the Flexowriter. The magnitude section of the word to be printed is divided into six parts of octal digits and is coded for reference to each key on the Flexowriter. There is no automatic tabbing after the printing of each word. Any desired editorial characters must be coded as any other alphabetic characters. This includes shifting up and down, color shifting, backspace, tabbing etc. It must be remembered, when printing in the alphabetic mode, that

the automatic carriage return tab stop must be removed to avoid returning the carriage while printing.

The number of digits which will be printed from the magnitude section of each word is controlled by the magnitude section of (m_2). This is effected by placing in (m_2) a binary one in the least significant binary digit position of the octal, decimal, or alphabetic digit position corresponding to the last digit which is to be typed from a single word. This feature affects only the magnitude section, and this applies to all three printing modes.

Examples:

```
21 1726 2100 0002
(1726) = 35 2000 2001 2003
(1727) = 26 2003 1325 2003
(2100) = 00 0000 0000 0000
```

Result: 35200020012003
26200313252003

```
21 1001 0204 0001
(1001) = 14 1761 2025 4400
(0204) = 03 0000 0000 0400
```

Result: 1001 3.14159

```
21 0000 0133 0010
(0000) = 00 3675 3274 4564
(0001) = 00 3666 3241 7560
(0002) = 00 7766 4171 6436
(0003) = 00 6232 4144 7464
(0004) = 00 3663 3245 7360
(0005) = 00 4475 4563 6436
(0006) = 00 6232 7776 7041
(0007) = 00 6654 1212 1234
(0133) = 10 0000 0000 0000
```

Result: The National Cash Register
Company

MODEL 126 MAGNETIC TAPE UNIT (Record)

Data may be read out of the computer and recorded on magnetic tape. This process is accomplished with a WRITE TAPE command (see SECTION III). As previously explained for the READ TAPE command, only eight words of one block are written with any one command. Therefore, to write in consecutive blocks a separate WRITE TAPE command must be given for each block. To write in both the odd and even numbered words in a block, the WRITE TAPE for the first section must be followed by a BLOCK SEARCH, and then another WRITE TAPE command for the second section.

IBM EQUIPMENT (Punching Cards)

The Model 102-A will operate with an IBM Gang Summary Punch, Type 523 or equivalent, to punch information in the computer on IBM cards. The IBM equipment is prepared for operation in the same manner described for the "readout" routine.

The cards to be punched are placed in the feed hopper face down, "12" or top edge first. If they are prepunched with data which are related to the data in the computer in a particular order, they must be arranged in the order in which the computer is prepared to punch them.

One card must be fed to the Summary Punch before the computer operations are started, so that the first card will be in position to be punched. As a new card is fed, the cards already in the Summary Punch advance to the next station. The Summary Punches, when used with the Model 102-A will perform only one operation, either read or punch. Two machines are required to have both

operations.

The Control Panel for the Summary Punch fits into the rack in front of the machine. This panel contains 80 inlet hubs for the punch magnets. These hubs are wired from the hubs for reading octal data and to the hubs for reading decimal data. The outlet hubs for punching octal data are identical to the inlet hubs for reading. They are grouped in the same fashion as for reading and refer to the same respective cells in the Buffer Register. Hub No. 20 emits a "12" punch signal when punching octal cards. Most often, octal cards will be punched for later re-entry into the computer. This hole can then be used to control the reading process.

The outlet hubs for punching decimal data are identical to the inlet hubs for reading. They are grouped in the same fashion as for reading and refer to the same respective cells in the Buffer Register.

SYSTEM OPERATING PROCEDURE

START (System ON)

1. Set MEMORY SAFE toggle switch on the Computer Control Panel in the OFF position.
2. Depress POWER ON pushbutton and wait until the POWER ON indicator light signals the power-on sequence is complete, indicator light ON.
3. Set MEMORY SAFE toggle switch to ON.
4. Depress CLEAR pushbutton and wait until IDLE indicator light comes ON.

Filling Operation (see previous description)

1. Set Fill switch to FILL.
2. Select Mode of operation: BASE 8 (octal) or BASE 10 (decimal) pushbuttons.

NOTE

If Model 126 Magnetic Tape Handling Unit(s) is being used with the computer, set operation switch on the unit's control panel to AUTO.

3. Fill the following tests daily before starting regular operation. (see SECTION VI):
 - a) Command Test Program
 - b) Erase Main Memory
 - c) Memory Sum, Shifted Sum
 - d) Minimum Access Test
 - e) Logical Voltage Check
 - f) Model 126, Search, Read, and Write Tests (if applicable)
 - g) IBM, Punch and Read Tests (if applicable)
4. Enter commands and data according to instructions under "Filling the Computer".
5. To specify a particular starting address to the control register, the desired octal address followed by an "f", followed by a hyphen (-), is typed on the Flexowriter.

Computation

1. Start computation by striking "s" key on Flexowriter or with pushbutton marked COMPUTE on the Computer Control Panel.
2. If the computer has been computing and stops from a HALT or FILL command, and the control register is not modified, computation will restart at command in cell following the HALT or FILL command.
3. Program control will bring computer to a stop after due process of computation.

Read Out Operation (see previous description)

1. Read out from the computer is controlled by the computer and

the instructions included in the program.

SHUTDOWN (System Off)

1. Set MEMORY SAFE toggle switch to OFF.
2. Set Flexowriter Power Switch to OFF (mounted on side of Flexowriter).
3. Depress POWER OFF pushbutton.

CAUTION

Do not remove power from the computer by turning off an external switch or breaker. Use the POWER OFF pushbutton first. Unless the external switch is thrown, a-c power will remain on the sequence relay rectifier and the fan and shutter motor. It is recommended that the a-c be left on for a brief period after the POWER OFF button has been depressed. This will permit the fan and shutters to operate to reduce the computer's internal temperature to room temperature.

FUNDAMENTALS OF PROGRAMMING GENERAL INFORMATION

When a problem involving numerical computation is delegated to a human operator for execution on a suitable desk type machine, it is necessary to furnish along with the problem a list of instructions and data, or "program." Similarly, if an automatic electronic computer is used to perform a task, it must also be supplied with a "program."

However, when dealing with an electronic computer, the problem of supplying a "program" becomes two-fold: (a) Programming, or preparation of the code; and (b) Proper insertion of the code into the machine. Item (b) has already been discussed in detail; therefore, the following paragraphs will deal only with the first of these problems.

The first phase of programming is planning. Once a suitable plan is formulated, it is best illustrated by means of a flow-chart. A flow-chart is a written schematic of the logical sequence of operations necessary for the completion of the given problem. In the second phase, each operation noted in the flow-chart is represented by one or more 102-A commands, which involves choosing suitable storage space in the memory and translating the program into numerically coded form.

To illustrate these essential phases of programming a simple example will be chosen. Consider, then, the problem of obtaining the difference between a given number, A, and the sum of a given set of ten numbers, and storing the result ready for printing. The development of a program which will supply a satisfactory answer to this problem will be explained in detail.

The procedure of obtaining a flow-chart is simplified by the use of formulas and algebraic symbols, whenever possible. For example, represent the ten given numbers by $c_1, c_2, \dots, c_{10}$, then the following equation can be written:

$$\text{Answer} = A - (c_1 + c_2 + \dots + c_{10})$$

Hence, the logical sequence of operations necessary to obtain the answer be-

comes more apparent. One of the many recommended forms of writing a flow-chart is the following:

Start 	
Add the 1st two numbers	
Add the 3rd number to the previous partial sum.	
.	
.	
.	
Add the 10th number to the previous partial sum.	
Subtract the sum obtained from A.	
Halt.	

The program is obtained by simply transforming the steps in the flow-chart into satisfactory 102-A commands. First, of course, the programmer must know the functional capabilities of the 102-A commands and how they must be coded.

Rather than attempt to introduce the reader to the entire command list at one time, tutorial examples such as this one will be selected to illustrate various commands. The entire command list, nevertheless, is explained in detail in SECTION III.

In this example, use is made of the commands which add and subtract numbers, and the command which halts computation. The word structure for these commands is as follows:

ADD - "ad" - code 35

I	m ₁	m ₂	m ₃
ad	address of augend	address of addend	address of sum

The ADD command will add the contents of memory cells m₁ and m₂ and store the sum with proper sign in memory

cell m₃.

SUBTRACT - "su" - code 36

I	m ₁	m ₂	m ₃
su	address of minuend	address of subtra- hend	address of difference

The SUBTRACT command will subtract the contents of memory cell m₂ from the contents of memory cell m₁, and store the difference with proper sign in memory cell m₃.

HALT - "ht" - code 22

I	m ₁	m ₂	m ₃
ht	xxxx	xxxx	xxxx

The HALT command will halt automatic computer operation and put the computer in a state of "rest". The m₁ and m₂ addresses are irrelevant, that is, they can be any four octal digit numbers. The m₃ address is irrelevant for the present purpose.

It is noteworthy, once again, that when the command word is actually coded it contains only the memory addresses locating these quantities--not the quantities themselves.

Once suitable storage space in the memory has been decided on for the commands needed in this program and for the eleven given numbers, the programmer will be ready to complete the code. Fortunately, the commands can be stored anywhere in the memory; the only restriction being the fact that, after choosing storage for the first command, subsequent commands must occupy successive memory cells, since normal computer operation is sequential. For this problem, assume the first command to be stored in cell 0100 and the eleven numerical constants, c₁, c₂, ..., c₁₀, A, in cells 0600 - 0612, respec-

tively. Also, working storage space (memory cells to store intermediate results) must be allotted. In this example cell 0700 was chosen to accumulate the answer.

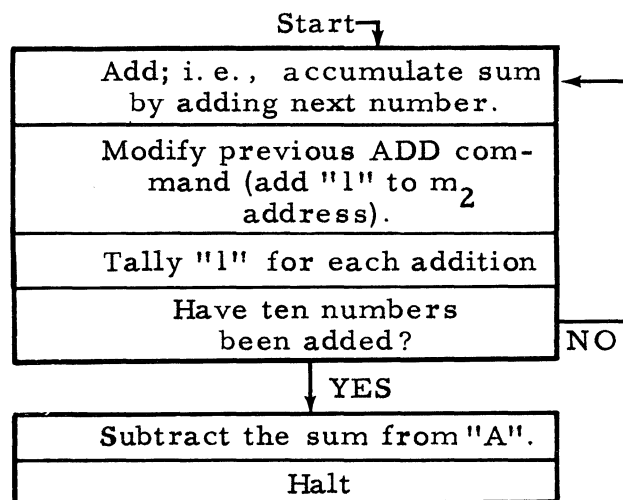
Referring to the flow-chart, the prescribed operations can now be replaced by the proper 102-A commands, coded with the selected memory addresses. The final code, shown in TABLE V-4 is in the standard format used by the programmer. Included, then, on every code sheet will be the word structure of the commands and numerical data, their respective memory addresses, and a "Remarks" column. The "Address" and "Remarks" columns are indispensable for coding subsequent commands after the first, during code checking, and as a future reference. Also, the "Address" column is needed when the program is filled into the computer.

Problems that warrant the use of an electronic computer are certainly much greater in magnitude and more profound than the previous example. The need for more subtle programming techniques is evident. For instance, if the previous problem had been one that involved many more additions, it would be impracticable to repeat the ADD command as was done. The development of a more efficient code, and one which will be applicable regardless of the number of additions, is not difficult once the programmer becomes familiar with certain fundamental programming techniques and with the versatility of the command list.

Since the objective in seeking a more efficient program is the elimination of repetitive ADD commands, consider the code just developed. Close examination of the nine successive ADD commands suggests that possibly one ADD command could be used for all the additions if its m_2 address could be successively increased by "1", and the ma-

chine's sequential mode of operation could be interrupted in order to repeat this modified ADD command. This cyclic procedure must also be controlled in some manner in order that it will terminate after the required number of additions have been made. To accomplish such a control, the machine must be made aware, at the end of each cycle, of the number of additions completed. Consider, then, the possibility of keeping a tally after each addition.

The essence of this procedure is more clearly illustrated by the following flow-chart:



The ability of the machine to modify its own commands and to make decisions is the crux of this procedure. In this example the prescribed logic can be fulfilled by means of the ADD and the TEST MAGNITUDE commands.

Consider first the modification of a command word, which is made possible by special use of the ADD command. A number can be added to a command word, and the instruction digits of the command word will be retained in the sum, if the address of the command word is named by m_1 of the ADD command. Consequently, the code will be written such that a word containing a "1" in its m_2 position will be added to

Address	I	m ₁	m ₂	m ₃	Remarks
0100	35	0600	0601	0700	$c_1 + c_2 \rightarrow 0700. (*)$
0101	35	0700	0602	0700	$(0700) + c_3 \rightarrow 0700. (**)$
0102	35	0700	0603	0700	$(0700) + c_4 \rightarrow 0700.$
0103	35	0700	0604	0700	$(0700) + c_5 \rightarrow 0700.$
0104	35	0700	0605	0700	$(0700) + c_6 \rightarrow 0700.$
0105	35	0700	0606	0700	$(0700) + c_7 \rightarrow 0700.$
0106	35	0700	0607	0700	$(0700) + c_8 \rightarrow 0700.$
0107	35	0700	0610	0700	$(0700) + c_9 \rightarrow 0700.$
0110	35	0700	0611	0700	$(0700) + c_{10} \rightarrow 0700.$
0111	36	0612	0700	0700	$A - (0070) \rightarrow 0700.$
0112	22	0000	0000	0000	Halt computation
0600	02	0000	0021	6532	$\left. \begin{array}{l} c_1 \\ c_2 \\ . \\ c_{10} \\ A \end{array} \right\} \text{Octal Numbers.}$
0601	00	0007	5063	1005	
.	.	.	.	.	
0611	02	0000	0110	7742	
0612	00	0000	0060	4306	
(*) - The arrow is used to symbolize "stored in": i. e., the sum $c_1 + c_2$ is stored in cell 0700." (**) - Parentheses used around the address of a cell indicates the contents of the cell: i. e., the sum $c_3 + (\text{contents of cell 0700})$ is stored in cell 0700.					

TABLE V-4. Program Format

the initial ADD command and recorded back in the same memory location. Thus, the initial ADD command will have been modified by augmenting its m_2 address by "1". Now, when the computer repeats this command (as indicated in the flow-chart) the next number will be added to the previous partial sum. Assuming the same memory storage for the program and the constants as in the previous example, the following will apply: Assume the initial ADD command to be

(0100) =

35	0700	0600	0700
----	------	------	------

anda "1" to be stored in the m_2 position of 0613

(0613) =

00	0000	0001	0000
----	------	------	------

Assume the second command in the program to be one which adds the two words previously

(0101) =

35	0100	0613	0100
----	------	------	------

The result of this command will be the following addition:

35	0700	0600	0700
----	------	------	------

00	0000	0001	0000
----	------	------	------

Sum

in (0100) =

35	0700	0601	0700
----	------	------	------

The reader is cautioned that this operation will be executed properly by the ADD command only if the command word is named by m_1 and the number by m_2 . If these addresses are interchanged in the ADD command, the machine will add the two words as two algebraic numbers, and will not retain the instruction digits of the command word.

Now to consider how to cause the computer to interrupt the normal sequence of operation and repeat a set of commands. Referring to the flow-chart, remember that the computer must tally after each addition. Therefore, it is necessary to add "1" to a tally storage

cell, and immediately ask the question: "Has the tally count reached 10?" This, then, is where the decision has to be made as to whether the computer will return to the ADD command, or continue in the program. If the answer is "no", the computer must cycle back and continue adding, but if the answer is "yes," the computer should be ready to continue.

The 102-A TEST MAGNITUDE command, whose word structure is shown below, is designed to ask such a question and act accordingly.

TEST MAGNITUDE - "tm" - code 34

I	m_1	m_2	m_3
tm	address of magnitude ₁	address of magnitude ₂	address of alternate command

This command actually causes the computer to compare the magnitude sections of the two memory cells named by its m_1 and m_2 addresses, and make the following decision: If the magnitude portion of cell m_1 is greater than that of m_2 , the address of the next command is automatically the m_3 address; otherwise, the computer will take the next command in the normal sequence.

In this problem a satisfactory decision will be made if the octal equivalent of "10" is pre-stored in the memory and its address is the m_1 portion of the "tm" command; also, the address of the variable tally count is placed in the m_2 portion; and m_3 contains the address of the initial ADD command. The code for the solution of this problem is shown in TABLE V-5.

This code could be used, then, to add as many numbers as desired, by merely changing the contents of the word, 0616, which acts as the gauge in the TEST MAGNITUDE command. The program constants in 0613-0616 would have to be stored elsewhere, however, if more

Address	I	m ₁	m ₂	m ₃	Remarks
0100	ad	35	0700	0600* 0700	Next number added to (0700) (0700) initially zero.
0101	ad	35	0100	0613 0100	Modify (0100); i. e., add "1" to m ₂ of 0100.
0102	ad	35	0614	0615 0615	Tally "1" in 0615.
0103	tm	34	0616	0615 0100	Has tally reached "10"? If not, next command 0100.
0104	su	36	0612	0700 0700	A - (0700) → 0700.
0105	ht	22	0000	0000 0000	Halt computation.
0600		02	0000	0021 6532	c ₁ .
0601		00	0007	5063 1005	c ₂ .
.		.	.	.	.
0611		02	0000	0110 7742	c ₁₀ .
0612		00	0000	0060 4306	A.
0613		00	0000	0001 0000	Modifier for 0101.
0614		00	0000	0000 0001	"1" for tallying.
0615		00	0000	0000 0000	Tally count storage, initially zero.
0616		00	0000	0000 0012	Octal equivalent of "10" as gauge for 0103.
0700		00	0000	0000 0000	Working storage, initially zero.
* Brackets are used in a command word to indicate which address in the word structure will be modified in the course of machine computation.					

TABLE V-5. Program Format

data were to be stored in channel 06.

The next example will introduce the reader to two more powerful machine tools, and a technique for using them.

Consider the problem of adding forty numbers which are stored in the memory in such a way that every storage word contains two of the given numbers. Assume each number to be six octal digits or less, and the two numbers "packed" in the same storage word to be of the same sign. The following diagram illustrates how each number will occupy half of the magnitude section of the storage word.

00	0506	73	11	4752
----	------	----	----	------

In this case the two numbers are +50673 and +114752.

In addition to the commands previously discussed it will be necessary in this problem for the programmer to be familiar with the SHIFT MAGNITUDE and EXTRACT commands. The word structure and function of these commands are as follows:

SHIFT MAGNITUDE - "sm" - code 30

I	m ₁	m ₂	m ₃
sm	address of word to be shifted	address of shift-control word	address of shifted result

Under the influence of this command, the machine will shift the magnitude section of a word a specified number of bit positions to the right or left; zeros will appear at either end of the magnitude section to replace digits that are shifted off the other end. The sign portion will remain unchanged. The m₁ address of the "sm" word structure identifies the word to be shifted; the m₃ address specifies the memory address in which to store the shifted result; the m₂ address designates the word whose

contents specify the direction and number of shifts. The contents of m₂ must be formed by the programmer in the following manner:

(m ₂) =	direction	number of bit positions to be shifted.
---------------------	-----------	--

The number of bit positions to be shifted is specified in the magnitude section as an octal number; the direction of shift is specified by the sign section: that is, a negative number will cause a right shift and a positive number will cause a left shift.

Also available, the SHIFT LOGICALLY command ("sl" - code 27) is identical to the SHIFT MAGNITUDE command except that the entire word is shifted.

EXTRACT - "ex" - code 32

I	m ₁	m ₂	m ₃
ex	address of word to be copied	address of extractor	address of cell altered by extract

The EXTRACT command permits selected bits from one word to be copied into the corresponding bit positions of another word. This operation functions in such a way that the binary digits of (m₁), which are in the same positions as the binary "ones" of (m₂), replace the corresponding binary digits of (m₃); and the bit positions of (m₃), corresponding to the binary "zeros" of (m₂) as a sieve in which "1" bits represent the holes. Bits of (m₁) are dropped through this sieve into (m₃).

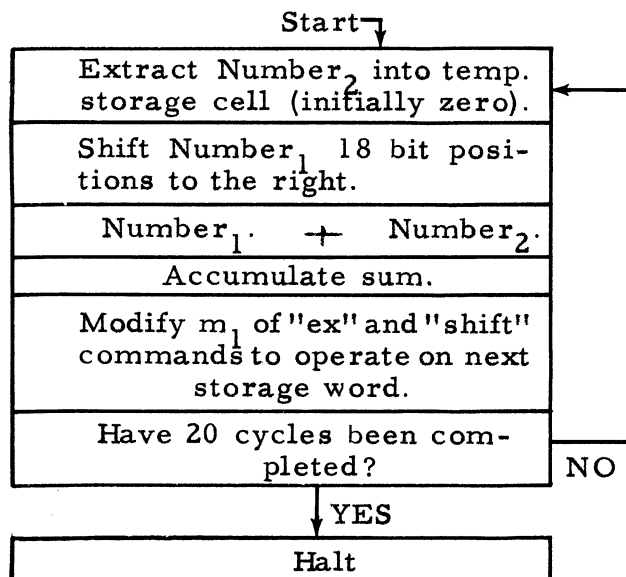
The availability of these two commands permits the following manipulations for the addition of the given forty numbers, and leaves the result stored in the memory. Consider, again, the plan of "packing" two numbers in a single word.

sign	Number ₁	Number ₂
------	---------------------	---------------------

and the steps necessary to "unpack" and add them:

1. Extract Number₂ into temporary storage.
2. Shift Number₁ completely to the right (18 bit positions).
3. Add the two numbers.
4. Accumulate sum.
5. Repeat this cycle until all forty numbers have been added (20 cycles).

Now the following flow-chart may be written:



The corresponding code which can now be written will introduce the programmer to the use of the buffer register, and a more advantageous way of making a decision than shown in the previous example (see TABLE V-6). Assume the first command to be stored in 0200; the given numbers in the twenty cells, 0300-0323; and the program constants in cells 0324-0327.

Use of buffer register cells in this program does not imply a new programming technique, but is intended to show the reader that they are used exactly

like main memory cells. The advantage in using cells 2000-2007 is that they are accessible to the computer eight times as fast, on the average, as are main memory cells.

The method used in this example to cause the computer to make the decision relative to cycling back is worthy of additional note. This is the method, perhaps, that most programmers would use, inasmuch as it eliminates tallying. The fact that the m_1 address of (0200) is being augmented by "1" each cycle provides the tally. Since the last two numbers to be added are stored in 0323, and since (0200) is modified at the end of the cycle, the contents of command word 0200 at the end of the last cycle will be:

(0200) =

32	0324	0324	0200
----	------	------	------

Hence, if cognizance is taken of this fact and a word whose magnitude section equals that noted above is pre-stored, the programmer will have a "gauge" word (or limiter) for the TEST MAGNITUDE command in 0205. The reader should recall now that the TEST MAGNITUDE command causes the computer to take its next command from the m_3 address only if the contents of m_1 are greater than the contents of m_2 . Consequently, the following word would be stored in 0327:

(0327) =

00	0324	0324	2000
----	------	------	------

Now, when the computer executes the "tm" command in cell 0206

(0206) =

34	0327	0200	0200
----	------	------	------

It will recognize equality between (0327) and (0200), and take its next command from 0207, which is the HALT command.

The next example is intended to instruct the reader in the use of the TEST FOR OVERFLOW MARKER command as a means of transferring computer control away from the normal sequential mode of operation.

Address	I	m ₁	m ₂	m ₃	Remarks	
0200	ex	32	0300	0324	2000	Number ₂ →2000.
0201	sm	30	0300	0325	2001	Number ₁ shifted to right and→2001.
0202	ad	35	2000	2001	2002	(Number ₁ + Number ₂)→2002.
0203	ad	35	2002	2003	2003	Accumulated sum (2002)→2003.
0204	ad	35	0200	0326	0200	Modify (0200); i.e. , add "1" to m ₁ of 0200.
0205	ad	35	0201	0326	0201	Modify (0201); i.e. , -add "1" to m ₁ of 0201.
0206	tm	34	0327	0200	0200	Have 20 cycles been completed: i. e. , does m ₁ of 0200 = 0324 yet?
0207	ht	22	0000	0000	0000	Halt.
0300		00	0506	7311	4752	} Given data.
0301		02	6362	4700	4430	
.		.	.	.	.	
0323		00	0004	0211	1252	
0324		02	0000	0077	7777	Extractor for 0200, extracts 6 octal digits and sign.
0325		02	0000	0000	0022	Shift constant for 0201; shift right, 18 bits.
0326		00	0001	0000	0000	Modifier for 0204, modifies (0200).
0327		00	0324	0324	2000	"Gauge" for test in 0205.
2003		00	0000	0000	0000	Working storage, initially zero.

TABLE V-6. Program Format

The word structure and function of this command are as follows:

TEST FOR OVERFLOW MARKER -
"to" - code 37

I	m ₁	m ₂	m ₃
to	address of word to be tested	xxxx	address of alternate command

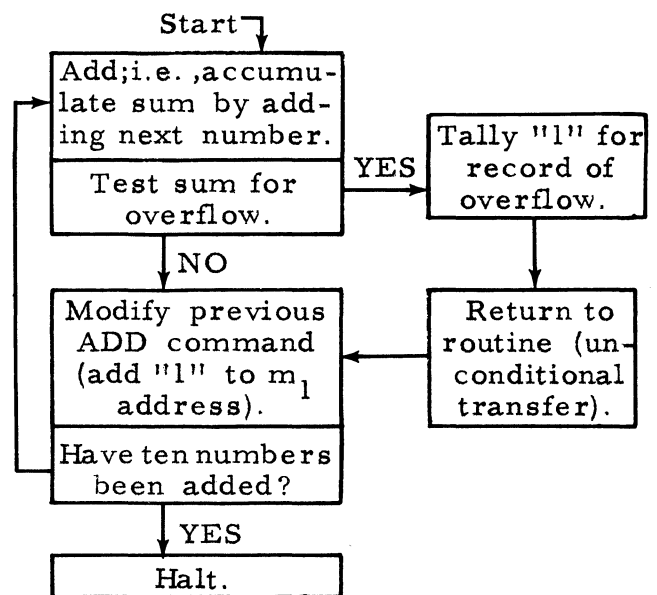
The "to" command causes the computer to test the contents of the word named by the m₁ address for the presence of a binary "one" in the overflow bit position. If a bit is present in this position, the machine will automatically take its next command from the address named in m₃; otherwise, it will continue in the normal sequence. The m₂ address is irrelevant. It is noteworthy that in this command the computer bases its decision merely on the presence of a "one" bit in the overflow position of the specified word, and does not concern itself with how it got there.

One of the principal uses of the "to" command is in conjunction with the ADD, SUBTRACT, or DIVIDE commands. The reader's attention is called to the automatic alarm signal built in the computer; that is, if an overflow bit is generated as a result of any of these commands the computer will automatically print the contents of the G register (cell 3000) on the Flexowriter and halt, unless the command in the program immediately following the command that caused the overflow is either "test for overflow marker" or "shift logically".

In many problems the programmer knows ahead of time that the magnitude of the data will not exceed the thirty-six bit capacity of a word; consequently, it is not necessary to incorporate any precautionary logic in the program. However, when it is not known ahead of time whether overflow will occur, it is necessary to include, in the program,

logic which will satisfactorily handle the situation and also prevent the computer from halting. The following example illustrates a typical method of using the "to" command for this purpose.

Assuming the possibility of overflow now, the following flow-chart shows how the programmer might handle this situation in a problem which requires the sum of ten given numbers:



Since the ADD command will be followed by a TEST FOR OVERFLOW MARKER command, the computer will not halt if overflow occurs, but will take its next command from a cell which will instruct it to tally "1" and unconditionally return to the routine so that the problem may continue. At the conclusion of the problem, the cell containing the tally of overflows would be the most significant part of the sum. Consider the ten numbers stored in cells 1100 - 1111, and the program beginning in 1200. The corresponding code is shown in TABLE V-7.

Emphasis is placed on commands, 1205-1206, which keep track of the number of overflows and return computer con-

trol to the routine of adding. Command 1206 is an unconditional transfer command. It is coded in such a way that the test will always "word" and computer control will always transfer to

1202. The reader is reminded that address 2100 supplies the programmer with a cell of all zeros; hence, any available non-zero program constant can be used as the m_1 reference in

Address	I	m_1	m_2	m_3	Remarks	
1200	ad	35	1100	2000	2000	Next number added to (200), initially zero. Is an overflow bit present in (2000) ?
1201	to	37	2000	2100*	1205	
1202	ad	35	1200	1207	1200	Modify (1200); i. e. , add "1" to m_1 of 1200.
1203	tm	34	1210	1200	1200	Have ten numbers been added? No, return to 1200.
1204	ht	22	0000	0000	0000	Halt computation
1205	ad	35	1211	2001	2001	"1" + (2001)→2001 - (overflow tally), initially zero.
1206	tm	34	1210	2100	1202	Return to 1202.
1207		00	0000	0000	0000	Modifier for 1202. "Gauge" for test in 1203 "1" for tally in 1205. } Numerical Data.
1210		00	1112	2000	2000	
1211		00	0000	0000	0001	
1100		00	6502	3450	0000	
1101		02	7300	2540	0300	
		.	.	.	.	
1111		00	0470	6600	0000	
2000		00	0000	0000	0000	Working storage, initially zero.
2001		00	0000	0000	0000	Used for tally, initially zero.
*Since the m_2 address in the "to" command is irrelevant, use 2100.						

TABLE V-7. Program Format

command 1206.

TECHNIQUES OF PROGRAMMING

The following paragraphs will be devoted to programs which will demonstrate the use of commands and programming techniques that heretofore have not been illustrated. Generally, these programs will be of a practical nature and can perhaps be used efficiently in actual problems that the programmer will run on the 102-A. Detailed explanations of the techniques used in each example will be given with the code. Before proceeding, the reader should familiarize himself with the information under "Binary and Octal Arithmetic" in the APPENDIX.

Inasmuch as decimal numbers must be converted to binary numbers before they can be operated upon in the 102-A, consider the program for the conversion of an integral decimal number to an equivalent binary number.

Example 1: Convert the decimal number, 89079, to a binary number. Assume, however, that the number is stored in a word in the computer as indicated:

0	0	0	0	0	0	8	9	0	7	9
---	---	---	---	---	---	---	---	---	---	---

The reader is reminded that this number is recorded in the memory as a sequence of binary coded decimal digits (4 bits per decimal digits) as shown below.

00	0000	0000	0000	0000	0000	1000
{ 1001 0000 0111 1001 }						

The solution of the problem will be more apparent if the given number is written in the following form:

$$89079 = 8 \times 10^4 + 9 \times 10^3 + 0 \times 10^2 + 7 \times 10^1 + 9 \times 1.$$

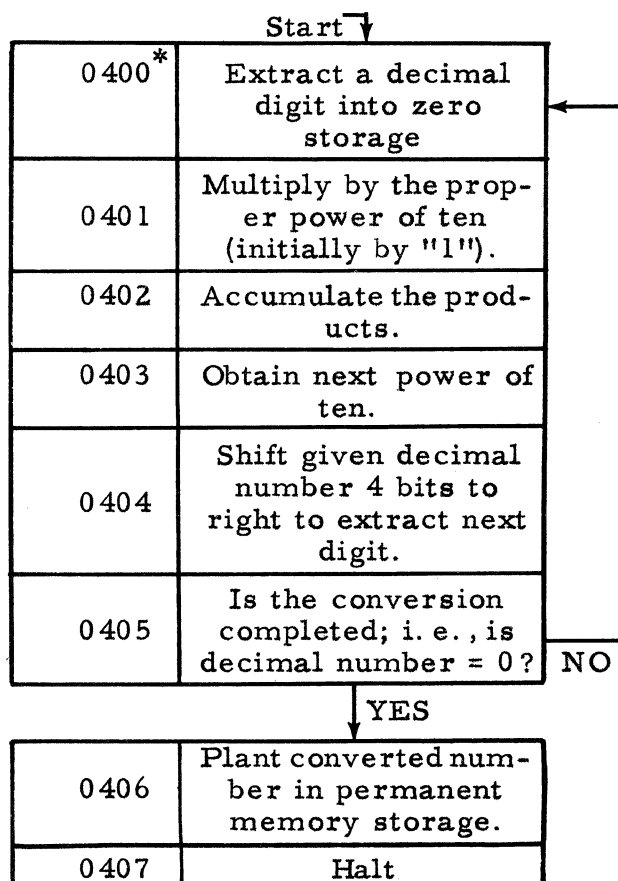
Since each decimal digit of the given number is represented within the machine by its equivalent four binary digits,

it is possible to store the binary equivalents of "1" and "10" in a word and then generate the binary equivalents of each term in the expanded form of the decimal number. Then, when these terms are added their sum would yield the equivalent binary number desired.

This is outlined below:

1. Extract a binary coded decimal digit.
2. Multiply by the proper power of ten which is expressed in binary form.
3. Add the products obtained in step 2 to form the desired binary number.

The following flow chart incorporates the steps outlined above.



*It is helpful to include addresses in the charts once the code has been written.

Memory storage was chosen as follows:

Decimal number to be converted in 0200; commands and program constants in 0400 - 0413; converted number will be stored in 0300; working storage in the buffer register. The program is coded in TABLE V-8 and a detailed description is included in the following paragraphs.

In command 0400 the reader is introduced to a technique which is equivalent to extracting bits from a selected word in the memory into a word of all zeros. This technique eliminates the necessity of first clearing the cell that the bits are to be copied into, but also requires that the cell being copied into

be a buffer cell, which in most cases is an advantage. Command, 32 0200 0410 2101, accomplishes this feat because of the 2101 address in m_3 . The reason is that the "extract" operation automatically requires the contents of m_3 to be read into the arithmetic unit, and after the "extract" operation is completed, the result is then automatically written into memory address m_3 . The "ex" command is the only 102-A command that utilizes the m_3 address for both reading and writing processes during its execution. Since the computer interprets addresses such as 210X (X = 0-7) differently when reading and writing, 2101 will be interpreted as 2100 during the reading process; hence,

Address	I	m_1	m_2	m_3	Remarks
0400	ex	32	0200	0410 2101	Next decimal digit $\rightarrow$ (clear) 2001.
0401	md	26	0411	2001 2002	(Decimal digit) x (power of ten) $\rightarrow$ 2002.
0402	ad	35	2004	2002 2004	Accumulated products $\rightarrow$ 2004 (initially 0).
0403	md	26	0411	0412 0411	Form next higher power of ten $\rightarrow$ 0411.
0404	sm	30	0200	0413 0200	Shift decimal number 4 bits to the right.
0405	tm	34	0200	2100 0400	Is decimal number = 0? No, return to 0400.
0406	ad	35	2004	2100 0300	Plant converted number in 0300.
0407	ht	22	2100	2100 0000	Halt computation.
0410		02	0000	0000 0017	Decimal digit and sign extractor for 0400.
0411		00	0000	0000 0001	"1" for units digit for 0403.
0412		00	0000	0000 0012	"10" for forming powers in 0403.
0413		02	0000	0000 0004	Shifter for decimal number for 0404.
2004		00	0000	0000 0000	Temporary storage for generating converted number.

TABLE V-8 Conversion of an Integral Decimal Number to an Equivalent Binary Number.

a word of all zeros will be read into the arithmetic unit, into which the desired bits will be extracted. However, when writing the result of the extraction into the memory, 2101 will be interpreted by the computer as 2001. It is noteworthy that in this procedure the original contents of 2001 are irrelevant.

Note that the extractor, (0410), contains a "1" in its sign bit position. Thus, this routine could be used to convert negative numbers as well as positive numbers. Each time a decimal digit would be extracted it would contain its respective sign bit; consequently, the subsequent products and sums would carry the proper sign.

Command 0401, the MULTIPLY DOUBLE LENGTH command will be discussed at length since it has not been illustrated in a program until now. This "md" command will multiply the decimal digit previously extracted into 2001 by the proper power of "10" stored in cell 0411, and write the two-word product in cells 2002 and 2003. Even though a two-word capacity is not needed for the product in any of the multiplications in this problem, the "md" has been used because of the nature of the problem and the type of multiplication (integers). Since mechanization of the 102-A "md" command is similar to that of a desk calculator or similar to manual multiplication, it is rather simple for the programmer to decide on how to position the factors in their respective cells in order that the binary point appear in the proper position in the product. In locating the binary point in the product, the programmer merely adds the binary places to the assumed binary point of each factor and this sum equals the number of binary places from the least significant end of the product. For example, during the first cycle of computation in this problem, the product of the contents of 0411 and 2001 is shown as follows (for convenience, decimal numbers are used throughout this

portion of the example):

(0411) =	0 0 0 0 0 0 0 0 0 1.
(2001) =	0 0 0 0 0 0 0 0 0 9.
product =	0 0 0 0 0 0 0 0 0 0 0 0
	(2003)
	0 0 0 0 0 0 0 0 9. 0 0 0 0 0 0 0 0 9.
	(2002) (zero placed)

The reader is reminded that in "md" the m_3 address, 2002, is that of the least significant half of the two-word product; also, the most significant half of the product, which is automatically recorded in the cell physically adjacent to 2002, will be zero because of the size and position of the factors in 0411 and 2001. When the programmer intends to utilize only one cell of the two-word "md" product, he often forgets to allow storage for the second cell. The programmer must remember to make certain, then, that 2003 is not being used as temporary storage to preserve a command or number to be used later in the program.

Command 0402 is intended to accumulate the sum of the products of the decimal digits and their respective powers of ten. The product of the decimal digit and its respective powers of ten, stored each cycle in 2002, is added to the sum of the products accumulated thus far in the routine in 2004. Initially, 2004 is filled with all zeros. Command 0403 is used to form the next higher power often needed when the next decimal digit is to be operated upon. Initially (0411), named as m_1 in this command, is filled equal to "1". After being used in command 0401, this command makes (0411) = "10." Then, during the next cycle (0411) is made = "100" etc.. Since this command also requires a two-word put away

for the result, it seems, at first glance, that the most significant half of the product (zero) would be written into 0412 and wipe out the constant multiplier, "10." However in the main memory, numerically consecutive cells are not physically adjacent. In this case, the next cell physically adjacent to 0411 is 0452, which is not being used in the program anyway.

The actual manner in which the sectors of physically adjacent main memory cells are numbered has been previously discussed in detail; however, for the present it should be sufficient for the reader to know that physically adjacent sectors are addressed octal forty-one apart.

Command 0404 causes the decimal number stored in 0200 to be shifted four bit positions to the right ("-4" in 0413) and recorded back to 0200. As a result of this shift the next significant decimal digit in the given number will be in the units position, ready for extraction at the beginning of the next cycle.

Command 0405 introduces a very common programming technique for causing the computer to make a decision. The essence of this technique is to test the magnitude of the decimal number, after the previous decimal digit has been utilized and the decimal number has been shifted by command 0404, against a word of all zeros. Until the decimal number becomes "zero" (completely shifted off the right end of 0200), the computer will recognize the magnitude of the decimal number (0200) as being larger than zero, (2100); and control will be transferred back to command 0400. When the decimal number in 0200 does become "zero", computer control will continue with the command in 0406.

The principle advantage of using this

technique is to generalize the routine for converting any number. The reader is reminded that the decision which this command causes the computer to make is based on whether the given decimal number is zero (all digits have been utilized), and not on the number of digits present in the given number.

Command 0406, the "ad" command, has been used to illustrate the most obvious way to transfer the contents of a cell to another memory location. Examination of the word structure in this command reveals that it is possible to add the converted number in 2003 to "zero" (2100), and record the result in 0300. Obviously, adding "zero" to (2003) will not change the contents of 2003.

Command 0407 halts computation and the computer is put in a state of "rest." Since m_1 and m_2 are irrelevant in the HALT command, 2100 is used.

Example 2: Convert the fractional decimal number, .73965, to a binary fraction. Assume the given number to be stored in the memory as indicated

0	0	7	3	9	6	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---

Writing the given number in its expanded form,

$$.73965 = \frac{7}{10} + \frac{3}{10^2} + \frac{9}{10^3} + \frac{6}{10^4} + \frac{5}{10^5}.$$

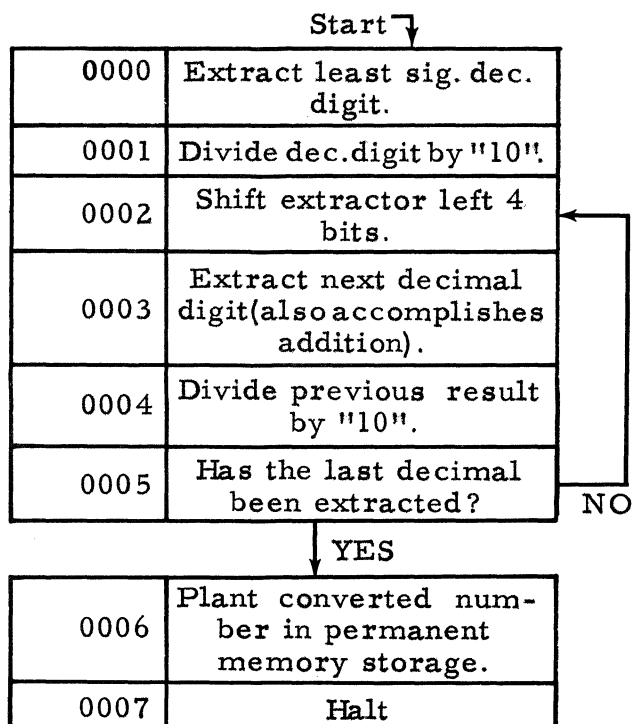
An equivalent form would be

$$.73965 = \frac{1}{10} \left\{ 7 + \frac{1}{10} \left[3 + \frac{1}{10} \left(9 + \frac{1}{10} \left(6 + \frac{5}{10} \right) \right) \right] \right\}$$

which suggests a definite plan for the program. For example, assume that the binary equivalent of decimal "10" (binary 1010) has been stored in a memory cell. Now, if the binary coded digit, 5 is extracted and divided by this pre-stored "10" the binary equivalent of .5 is obtained. Next, extract and add the binary coded digit, 6, and obtain the binary equivalent of 6.5. Dividing by

"10" again, adding "9", dividing by "10", etc., finally obtains the binary equivalent of .73965.

The flow-chart would appear as follows:



The corresponding code and selected memory storage is shown in TABLE V-9. A detailed description of the code is included in the following paragraphs.

Command 0000 extracts the least significant decimal digit. Since the extractor in 0010 contains four binary ones in the fifth digit position, the least significant decimal digit will be extracted into 2000. Also, the binary one in the sign bit position of 0010 will extract the proper sign bit with the decimal digit. Once again the technique of addressing m_3 as 210X has been used to insure extraction into a cell of all zeros.

Command 0001 divides the decimal digit

by "10." Consider the binary contents of the dividend and divisor immediately after the extraction of the decimal digit "5". Decimal points are indicated in

(2000) = 00 0000 0000 0000 0000 0000
Dividend 0101. 0000 0000 0000 0000

(0010) = 000 010 000 000 000 000 000
011 110 000 ...

(0011) = 00 0000 1010. 0000 0000 0000
Divisor 0000 0000 0000 0000 0000

the word structure to aid in the discussion. The reader is reminded that the binary equivalent of "10" (divisor) is 1010, but indicated on the code sheet as an octal number, 00 5000...

This command causes the binary equivalent of the decimal digit "5", extracted into 2000, to be divided by the binary equivalent of decimal digit "10", and the quotient to be rounded and stored in 2001. Emphasis must be placed on the choice of position of the divisor in this example. Actually, the dividend and divisor are being treated as integers in this program; but, placement of the divisor within a word depends on where the programmer wants to consider the decimal point in the quotient and how mechanized division in the 102-A functions.

The choice of placing the divisor at the extreme left end of the word was based on the fact that the next phase of computation required in this program is the addition of the next decimal digit. Since the EXTRACT command has been chosen as the means of adding the next decimal digit, the decimal point in the quotient (. 5) should be located four decimal digit

Address	I	m ₁	m ₂	m ₃	Remarks
0000	ex	32	0100	0010 2100	Lst. sig. decimal digit → clear 2000.
0001	dr	23	2000	0011 2001	(2000) divided by "10" → 2001.
0002	sm	30	0010	0012 0010	Shift (0010) (extractor) left 4 bits.
0003	ex	32	0100	0010 2001	Extract and add next dec. digit into 2001.
0004	dr	23	2001	0011 2002	Divide number being generated, (2001), by "10".
0005	tm	34	0013	0010 0002	Is extractor in most sig. dec. digit pos. ?
0006	ad	35	2002	2100 0200	Plant converted number in 0200.
0007	ht	22	0000	0000 0000	Halt computation.
0010		02	0000	0360 0000	Initial form of extractor; 36=four bin. 1's.
0011		00	5000	0000 0000	Divisor, 0001 and 0004; equiv. to bin. 1010.
0012		00	0000	0000 0004	Shifter for 0002.
0013		00	7400	0000 0000	Gauge for 0005.

TABLE V-9 Conversion of a Fractional Decimal Number to an Equivalent Binary Number.

positions to the right of the most significant end of the word.

(2001) =

0	0	0	0	0	0	.	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

← position into which
next digit will be
extracted.

Since the 102-A automatically considers the binary point in the dividend and divisor to be located in the same position during division and forms the quotient in such a way that its point is assumed to be located between sign and magnitude sections, actual location of the point in the divisor to the right or left of that in the dividend is equivalent to shifting the point in the quotient an equal number of positions in the opposite direction. Hence, in this example, each

time the next decimal digit is extracted into the previous quotient the next division by "10" will theoretically place the point four bit positions to the left.

For example, extract the first digit.

0	0	0	0	0	0	5	.	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

Divide by "10"

0	0	0	0	0	0	.	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

Extract next decimal digit.

0	0	0	0	0	6	.	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

Divide by "10".

0	0	0	0	0	.	6	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

Extract next decimal digit.

0	0	0	0	9	.	6	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

Divide by "10"

0	0	.	9	6	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---

Continuing this process will produce the desired binary equivalent with the point as indicated

0	0	.	7	3	9	6	5	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

An alternate method for locating the binary point in a quotient is to keep track of it by powers of "2". Consider the binary point in any dividend and divisor as being located at the machine point (between sign and magnitude sections), and represent the number of positions the true point is to the right of the machine point as a negative power of "2". Manual division of the dividend and divisor represented in this manner will yield the position of the binary point in the quotient as the result of the division commands. For example, consider the first division in this problem:

$$\frac{5.}{10.} = \frac{5 \times 2^{-20}}{10 \times 2^{-4}}$$

(where 2^{-20} = 5 decimal digits or 20 bit positions to the right)

(where 2^{-4} = 1 decimal digit or 4 bit positions to the right)

$$\text{result} = .5 \times 2^{-16}$$

(where 2^{-16} = 4 decimal digits or 16 bit positions to the right)

Command 0002 causes the contents of 0010 (extractor) to be shifted four bit positions to the left and the result written back in 0010.

This shift results from the "+4" code stored in 0012. As a result of this command, the "extractor" is in a position to extract the next decimal digit when needed.

The EXTRACT command (0003), formed by the "sm" command in 0002, is used to extract all subsequent decimal digits,

after the first, into the quotient in 2001 obtained from the previous cycle.

Command 0004, the "dr" command, performs the subsequent divisions by "10", after the first, and stores the quotient in 2002.

Command 0005 enables the computer to decide if the conversion has been completed. The conversion will be completed, of course, when the last decimal digit has been extracted and the result divided by "10". The technique used here is to store a gauge word which will be equivalent to the configuration of the magnitude section of the "extractor" when the last decimal digit has been extracted. This gauge word is then continually tested against the extractor. When the extractor equals the gauge word the computer will discontinue cycling. For this reason 0013, the address of the gauge word, is located in m_1 and the variable extractor, 0010, in m_2 in this "tm" command. Since this command only compares the magnitude sections of 0013 and 0010, a "1" has not been stored in the sign bit position of 0013 as was done in 0010. Indicated below is the binary configuration of the gauge word.

0	0	7	4	0	0
(0013) = 000 000 111 100 000 000					
0	0	0	0		
000 000 000 000...					

The initial configuration of the extractor was shown earlier in this discussion. After it is shifted left four bit positions (1 decimal digit) the first time by command 0002 it will take on the following form

000	010	000	000	000	000
(0010) = 000 010 000 000 000 000					
111 100 000 000...					

The magnitude section of (0013) is greater than the magnitude section of (0010)

and the computer will cycle back to command 0002 in order to operate on the next decimal digit. Finally, after 0010 has been shifted and used to extract the last decimal digit, it will contain the following configuration.

(0010) =

000	010	111	100	000	000
-----	-----	-----	-----	-----	-----

000	000	000	000	...
-----	-----	-----	-----	-----

Now, when the computer makes the comparison it will find that the magnitude section of 0013 is not greater than that of 0010 (equality is not considered greater than) and will continue with the next command in numerical sequence (0006).

In command 0006, the converted number generated in 2002 is transferred to the permanent memory location, 0200; that is, its contents are added to zero and recorded in 0200.

Command 0007, the "ht" command, stops automatic computer operation and puts it in a state of "rest."

In the previous examples the entire program has been outlined in detail. In the remaining examples in this section, detailed explanations will be given relative to the concepts and techniques which require emphasis. It is assumed that the reader will scrutinize the programs carefully and verify the coded commands which are used.

Example 3: Convert a fractional binary number to a fractional binary coded decimal number. Assume the binary number to be stored in a word in the computer as follows:

$$0 \pm .b_1 b_2 b_3 \dots b_{36}$$

(The b_i 's represent the binary digits of the given number.)

Represent the desired decimal result as, $.d_1 d_2 d_3 \dots d_9$, and the following equality is derived,

$$\frac{b_1}{2} + \frac{b_2}{2^2} + \frac{b_3}{2^3} + \dots + \frac{b_{36}}{2^{36}} = \frac{d_1}{10} + \frac{d_2}{10^2} + \frac{d_3}{10^3} + \dots + \frac{d_9}{10^9}$$

The problem, of course, is to determine the binary coded decimal digits, d_1, d_2 , etc., within a word in the computer ready for decimal print-out.

Consider the following plan: If both sides of the previous equation are multiplied by "ten" in their respective number systems, the result will be two equal mixed numbers.

For example, the right hand side becomes

$$d_1 + \frac{d_2}{10} + \frac{d_3}{10^2} + \dots + \frac{d_9}{10^8}$$

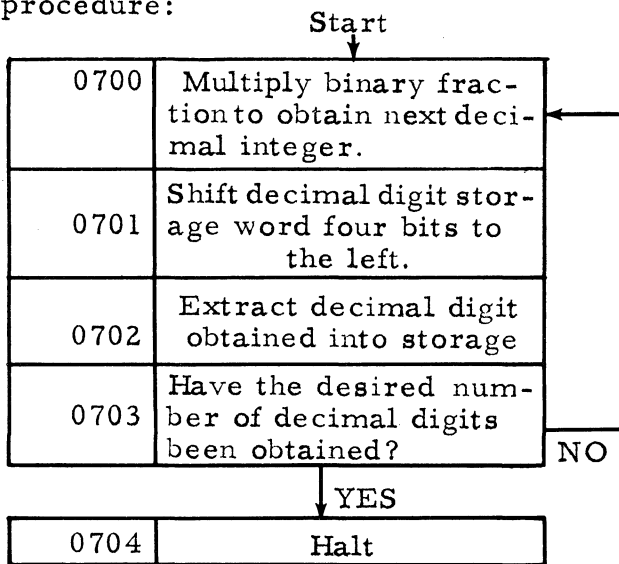
Since the whole parts and the fractional parts of two equal numbers must be equal respectively, the integer, (d_1), will equal the integral part of the binary product on the left side. Consider, now, only the fractional parts of each side. Successive multiplication by "ten" of the fractional part of each product would yield the binary coded decimal digits, d_2, d_3 , etc..

For the purpose of writing a satisfactory flow-chart, consider the following detailed plan:

1. Multiply the given binary fraction by the binary equivalent of "ten".
2. Take the integral part of the product, d_1 , and store in temporary storage.

3. Multiply the fractional part of the product by "ten" again.
4. Shift the word, which stores the first decimal integer, four bit positions to the left and store the second decimal digit obtained from the previous product.
5. Continue this process of multiplying, shifting the decimal storage word and sorting the next decimal digit, until the desired accuracy is obtained or until the nine decimal digit capacity of the storage word is used.

The following flow-chart illustrates this procedure:



The corresponding code is shown in TABLE V-10.

In addition to illustrating a method for converting a binary fraction to an equivalent decimal fraction, this example is intended to illustrate a subtle way of using the combination of the "sl" and "to" commands to tally and to make a decision. Use of these commands for this purpose in this problem is dependent on the initial configuration of the program constant stored in cell 0705.

The technique will be more apparent if this octal constant is written in binary form:

00 0421 0421 0420

(0705) = 000000 0001 0001 0001 0001
0001 0001 0001 0001 0000

Now, when this entire word is shifted left four bit positions by the SHIFT LOGICAL command (0701), a "1" will appear in the overflow bit position eight times in succession. Consequently, when command 0703 tests this word for overflow, the computer will transfer control back to command 0700. However, the ninth time this word is shifted, and after the ninth decimal digit has been extracted into it, the overflow bit position will contain a "0" and the computer will take the next command in normal sequence, 0704.

Address	I	m ₁	m ₂	m ₃	Remarks
0700	md	26	2000	0710 2000	Product: Integer → 2001; fraction → 2000.
0701	sl	27	0705	0706 0705	Storage word shifted 4 bits left.
0702	ex	32	2001	0707 0705	Extract decimal digit → 0705.
0703	to	37	0705	2100 0700	If overflow bit, then → 0700.
0704	ht	22	2100	2100 f	Halt.
0705		00	0421	0421 0420	Storage for decimal digits.
0706		00	f	f 0004	Shifter for 0701.
0707		02	f	f 0017	Extractor for 0702.
0710		00	f	f 0012	Constant multiplier, "10".
NOTE: "f" is conveniently used on the code sheet to indicate four octal zeros.					

TABLE V-10 Conversion of a Fractional Binary Number
to a Fractional Decimal Number.

Maintenance

INTRODUCTION

MAINTENANCE PROGRAM

A complete preventive maintenance program has been developed for the Model 102-A computing system. The program is designed to perform a twofold function. First, by periodically running the series of test programs with known results, and varying selected parameters (voltages, clipping levels, etc.), the over-all operation of the computer and auxiliary equipment can be evaluated to determine the range of variation over which the system operates reliably (marginal checking). This technique permits the detection of incipient failure of components in the system before their deterioration reaches the level of failure during actual computation.

Second, the systematic program of measurement and adjustment of all variables, which are subject to change as a result of normal component aging, serves to increase the useful life of components in the computer. Such variables include power supply voltages, memory amplifier gains, etc.

In addition, a certain amount of scheduled maintenance of electromechanical devices within the system will decrease the probability of their malfunction during actual computation. Such devices include the Flexowriter, sequence relays, controls, indicators, and motors.

It is important to note that in any case where a preventive maintenance test indicates a weakness in the system, the component responsible for such indica-

tion should be located and repaired immediately.

If the system is permitted to lapse into marginal operation condition through irresponsible preventive maintenance, multiple component failures may occur which will be very difficult to locate and will seriously reduce the percentage of usable time for the computer.

Maintenance personnel should keep a firm schedule of preventive maintenance operations (see SECTION II, "System's Records"), and record all preventive maintenance time and a description of services performed in the log book. Customer Service engineers of The National Cash Register Company, cooperate to keep these records up to date by filing service reports covering each preventive maintenance call in detail.

NOTE

Before attempting any of the internal adjustments or measurements described in this section, maintenance personnel should become thoroughly familiar with the safety measures listed under "Precautions."

PRECAUTIONS

All circuits on the tube chassis and diode boards are electrically direct coupled. Accidental grounding or contact between any two terminals may cause considerable damage.

Avoid using tools, metallic objects, or clip leads on diode boards and tube chassis with dc on.

Turn dc off to remove or replace any plug-in unit.

All voltage measurements should be taken with a 20,000 ohms-per-volt meter or vtvm (+1 percent calibration is required).

Never turn off the Stabiline switch.

Do not remove power from the computer by turning off an external switch or breaker. Use the "power off" button first, then throw external switch if desired.

There is always power applied to the sequence relay (+90V) rectifier, even with "power off". Use external switch or pull main plug to work on system. Also, the Flexowriter should be turned off, since it supplies voltage to some of the input relays.

Never close a current measurement switch without first connecting a current meter into the circuit.

Make no adjustments to magnetic heads with the drum moving. Look at the shaft to be sure. (exception: fine tangential adjustment with amplitude monitored, as described under adjustment of "Recirculating Registers.")

If the unpleasant odor of oxidizing selenium is observed, shut off all power immediately, ventilate and vacate the area. Selenium in gaseous compounds is poisonous. Avoid working near the corners of the large selenium rectifiers. A scratch from a chipped rectifier plate could be fatal.

Any time the end panels are removed from the computer for more than a brief adjustment, the tube chassis should be

swung out and air from 16" electric fans should be directed against the tubes, in order to insure adequate cooling.

If the information in the main memory is important, the MEMORYSAFE switch should be thrown whenever any measurements or adjustments are made with the dc on.

After any failure which might have turned a record flip-flop on for one or more word times, it is recommended that the memory channels be erased to remove any P_2O_{13} pulses. The "Erase Main Memory" program in the APPENDIX may be used for this purpose.

MAINTENANCE EQUIPMENT

Spare Parts

A recommended list of spare parts has been assembled and should be maintained in stock to adequately service a typical 102-A computing system (Model 102-A and one or two Model 126 units). The list includes necessary plug-in units, tubes, electronic components (capacitors, diodes, resistors, etc.), switches, relays, and tools. (see APPENDIX, G.)

Test Equipment

The following recommended equipment is sufficient for complete servicing of a Model 102-A installation:

Oscilloscope

Vertical Bandwidth: DC to 5 megacycles.

Maximum Vertical Sensitivity: DC and AC, 0.01v or better per major vertical division. AC only, 0.01v or better per major vertical division.

Sweep: Triggered and recurrent.

Time-Base Ranges: 0.1 microsecond to 5 seconds per major horizontal division.

Amplitude Calibrator: Internal, square-wave; for peak-to-peak voltage measurements

Volt-Ohmmeter

DC Voltage ranges: 0-2. 5-1050-250-1000-5000v at 20,000 ohms per volt.

AC Voltage Ranges: 0-2. 5-10-50-250-1000-5000v at 1000 ohms per volt.

Current Ranges: 0-100 μ a, 0-10-100-500 ma, 0-10 amp. dc.

Resistance Ranges: 0-2000-200,000 ohms, 0-20 megohms.

Vacuum-Tube Voltmeter

Case: Must be insulated, since vtvm will be operated completely above ground in some computer tests.

AC and DC Voltage Ranges: 0-1. 2-12-60-300-1200v.

Input Resistance on DCV Ranges: 10 megohms constant, plus 1 megohm in dc probe handle.

Input Impedance on ACV Ranges: 275,000 ohms, 200 μ fd.

Resistance Ranges: 0-1000-100,000 ohms, 0-1 -- 10-1000 megohms.

Tube Tester

Transconductance Range: Direct-reading scales to 15,000 micromhos.

Additional Functions: Short-circuit test, noise test, gas test.

Computer Test Set: (Manufactured by the National Cash Register Company Electronics Division)

Model 388: For testing Alden-type plug-in units, diode characteristics, and emission and cutoff of vacuum tubes.

Model 389: Power Supply for Model 388.

PREVENTIVE MAINTENANCE SCHEDULE

DAILY

Operational Testing (Marginal Checking)

Command Test Program
Minimum Access Test

WEEKLY

Logical Voltage Check
Preventive Maintenance (using "Preventive Maintenance Check List" sheet.)

Operational Testing (Marginal Checking)

Reliability Program
Minimum Access Test
Tape Unit Read-Write Test
IBM Test

BIWEEKLY

Flexowriter Maintenance (using "Preventive Maintenance Check List" sheet.)

MONTHLY

Recirculating Register Adjustment (if required)
Main Memory Adjustment
Power Supply Ripple Check
Miscellaneous Maintenance (Air Filters, Relays, etc.)
Flexowriter

QUARTERLY

Complete Flexowriter Overhaul
Mechanical Inspection
Clock and Word Channel Adjustment

Preventive Maintenance Check List

The Preventive Maintenance Check List provides for testing of all plug-ins, tubes, diodes, and Flexowriter maintenance according to standard proced-

ures and on a regular schedule so that each component is tested at intervals within its predictable life. Using this Check List, approximately 15 weeks are required to complete a check on all components in the Model 102-A and Model 126, except logical diodes. All logical diodes are checked in 30 weeks.

OPERATIONAL TESTING

RELIABILITY PROGRAM

The reliability program is a comprehensive routine which tests each of the commands in the computer (except IBM and Magnetic Tape commands) under a wide variety of numerical conditions. Coding for the reliability program is given under "Reliability Program" in APPENDIX E.

NOTE

The memory should be cleared by running the Erase Main Memory subroutine at the start of the program; then, before filling the rest of the reliability routine, press the CLEAR button so that the memory check sums will be correct.

Filling the complete program takes about five minutes using IBM cards, or thirty minutes using Flexowriter tape.

The program is divided into three parts. Part I is an independent section of the program and may be used without filling the rest of the program. When Part I is used alone, computation should be started in 0001.

With the entire routine in the memory (Parts I, II, and III), computation should be started in 0000. Test Switches 2010 and 2100 should be ON, and the other two OFF. Test Switch 2100 controls the repetition of the test; if it is OFF, the computer will halt after one run (6 mins. 45 secs.). Test Switch 2010 controls the operation of Part II. If it is OFF, the computer will do Part III and Part I only.

After the first time through the routine, the computer should type out the following for each run of the test:

```
0626 35210021002100
2100 00000000000000
2100 00000000000000
2100 00000000000000
1774 35033611044332
1776 00365177127403
```

The alphabetic mode uses every Flexowriter key. (Judge how quickly it "duz" it!)

Any other type-out indicates a computer error.

Part I

Part I is composed of fourteen small routines which test all of the arithmetic and logical commands. Any error which might be made by the computer will be indicated by a type-out.

Time: 4-1/2 minutes.

- 1.) Commands start - 0001
 Commands end - 0010
 Constants (bl) - 0022, 0024, 0030
 The test adds the octal integers from 1 to 1777 and checks against the known sum.
 2000 1 (to augment the integer)
 2001
 2002 limiter (2000)
 2003 computed sum
 2004 correct sum (1777000)
 2005
 2006 integer (augmented by one each time)
 2007

Form of error print-out:

```
00000000000001
test number
```

```
XXXXXXXXXXXXXXXXX
known sum minus
computed sum
```

```
XXXXXXXXXXXXXXXXX
computed sum
```

- 2.) Commands start - 0011
 Commands end - 0032
 Constants (bl) - 0021, 0023-0025

The test adds the square of the octal integers from 1 to 777 and checks against a known sum.

2000
 2001 1
 2002 computed sum
 2003 limiter (1000)
 2004 integer (augmented by 1 each time)
 2005 correct sum (252125400)
 2006 square of integer (least significant half)
 2007 zero, most significant half of product.

Form of error print-out:

00000000000002
 test number

0XXXXXXXXXXXXXXXXX
 known sum minus
 computed sum

0XXXXXXXXXXXXXXXXX
 computed sum

- 3.) Commands start - 0033
 Commands end - 0044
 Constants (bl) - 0071, 0073, 0075
 The test adds the cube of the octal integers from 1 to 577 and checks against a known sum.
 2000 zero (most significant half of product)
 2001 1
 2002 computed sum
 2003 limiter (600)
 2004 integer (augmented by one at each step)
 2005 correct sum (50224110000)
 2006 square (least significant half of product)
 2007 zero; cube (least significant half of product)

Form of error print-out:

00000000000003
 test number

0XXXXXXXXXXXXXXXXX
 correct sum minus
 computed sum

0XXXXXXXXXXXXXXXXX
 computed sum

- 4.) Commands start - 0045
 Commands end - 0051
 Constants (bl) - 0072, 0074, 0076
 The test subtracts a one in the right-hand position from a number containing all sevens, and then shifts the magnitude of the (1) one binary place left, and subtracts again. It does this 36 times. The correct answer is zero (minuend).
- 2000
 2001
 2002 minuend
 (originally 00777777777777)
 2003
 2004 1 (to shift left one place)
 2005
 2006 subtrahend (originally 1)
 2007

Form of error print-out:

00000000000004
 test number

0XXXXXXXXXXXXXXXXX
 minuend (should be zero)

- 5.) Commands start - 0101
 Commands end - 0107
 Constants (bl) - 0072, 0074, 0076
 The test subtracts a negative one in the right-hand position from a negative number containing all sevens, and then shifts the mag-

nitude of the (-1) one binary place left, and subtracts again. It does this 36 times.

The correct answer is zero (minuend).

2000
2001
2002 minuend
(originally 02777777777777)
2003
2004 1 (to shift left one place)
2005
2006 subtrahend
(originally 02000000000001)
2007

Form of error print-out:

00000000000005
test number

0XXXXXXXXXXXXXX
minuend (should be zero)

6.) Commands start - 0110
Commands end - 0117
Constants (bl) - 0072, 0074, 0076

The test subtracts a one in the right-hand position from a negative number containing all sevens, then shifts the (1) logically one place left, and subtracts again. It does this 36 times and then checks the minuend. The correct answer is 03777777777776.

See the above buffer contents.

Form of error print-out:

00000000000006
test number

0XXXXXXXXXXXXXX
minuend

7.) Commands start - 0120
Commands end - 0130
Constants (bl) - 0072, 0074, 0076

The test subtracts a negative one in the right-hand position from a number containing all sevens, ignores the overflow, then shifts the magnitude of the (-1) one binary place left. It does this 36 times and then checks the minuend. The correct answer is: 01777777777776.

See the above buffer contents.

Form of error print-out:

00000000000007
test number

0XXXXXXXXXXXXXX
minuend

8.) Commands start - 0211
Commands end - 0214
Constants - 0457, 0474

This tests the addition of commands. Command 0211 adds a command into 0212 which adds a command into 0213 which clears 0213. Finally 0214 clears 0212. A failure is indicated by an error print-out.

9.) Commands start - 0300
Commands end - 0351
Constants - 0314-0337, 0353-0376

This test buffer-loads eight numbers and then extracts into the eight numbers to form the octal digits which it prints alphabetically as the following sentence:

The alphabetic mode uses every Flexowriter Key. It then buffer-loads eight more numbers, ex-

tracts into them, and prints the following, with the word key and the period following it in the physical position shown:
(Judge how quickly it "duz" it!)

The form of error print-out is failure to print the following two lines:

The alphabetic mode uses every Flexowriter Key.
(Judge how quickly it "duz" it)

- 10.) Commands start - 0146
Commands end - 0155
Constants (bl) - 0161, 0165

The test scale-factors N, and shifts logically the scaled N the number of places it was scale-factored. Then it shifts the magnitude of N one place right and repeats until N is zero. Finally, it checks the amount of the last shift.

2000 N (after scale factor)
2001 N
(originally 00777777777777)
2002
2003
2004
2005 -1
2006
2007 number of shifts to scale factor N

The form of error print-out:

00000000000010
test number

XXXXXXXXXXXXXXXXXX
number of shifts
(should be -43)

XXXXXXXXXXXXXXXXXX
N after scaling
(should be zero)

- 11.) Commands start - 0166
Commands end - 0170

Constants - 0156, 0175

This test scale-factors zero.

The form of error print-out:

00000000000011
test number
XXXXXXXXXXXXXXXXXX
number of shifts
(should be -100)

XXXXXXXXXXXXXXXXXX
zero (after scaling)

- 12.) Commands start - 0177
Commands end - 0210
Constants - 0221, 0223, 0225, 0021, 0222

This test multiplies and rounds $A \times R$ and adds the product to the sum. Then it multiplies $R(R) = (R)^2$, and multiplies this product by A. It repeats this sixteen times, and checks the sum against a known answer.
(00223305225450).

2000 $A(R)^P$
2001 $A = 1/2$
2002 R^P
2003 $R = -3/4$
2004 computed sum
(originally $1/2$)
2005 limiter (20)
2006
2007

The form of error print-out:

00000000000012
test number

XXXXXXXXXXXXXXXXXX
known sum minus
computed sum

XXXXXXXXXXXXXXXXXX
computed sum

- 13.) Commands start - 0215
 Commands end - 0253
 Constants (bl) 0230, 0232

This test divides and rounds B by 7, and stores the answer in the buffer. It then augments B by 1, modifies the divide command, and returns. It does this six times and then checks the quotients.

2000 lff, 6/7 (006666666666667)
 2001 B (originally lff)
 2002 7ff
 2003 1/7 (001111111111111)
 2004 2/7 (002222222222222)
 2005 3/7 (003333333333333)
 2006 4/7 (004444444444445)
 2007 5/7 (005555555555556)

The form of error print-out:

00000000000013

test number

XXXXXXXXXXXXXXXX

check sum (should be zero)

- 14.) Commands start - 0254
 Commands end - 0272
 Constants (bl) 0273, 0275, 0277, 0021

This test divides double B/7, and multiplies double the quotient by 7. It then adds the least significant half of the product to the remainder. If there is an overflow, it adds 1 to the most significant half of the product. Then it checks to see that B is equal to the product. It augments B by 1 and repeats until B is 7.

2000 remainder
 2001 quotient, product (least significant), sum
 2002 product (most significant) should equal B
 2003 7000ff (divisor)

2004
 2005 B (originally 10ff)
 2006 B minus the product
 2007 lff (augmenting B)

The form of error print-out:

00000000000014

test number

XXXXXXXXXXXXXXXX

sum (should be zero)

XXXXXXXXXXXXXXXX

product (should be equal to B)

Part II

Part II consists of a series of "minimum accessed" commands which pick up the operands at the earliest or latest possible time, and refers to every channel in the memory. The error print-out will type the address and the command that failed, and the address and the operands of that command. A failure here is illustrated by the following type-out:

0607 33135216151574
 1352 00000000000001
 1615 00000000000000
 1574 33013716021261

This indicates that command 0607 failed, and since (1352) = 1, and (1615) = 0, the control should have jumped to 1574, but instead the next command came from 0610 which is an entry to the print routine.

Time: 8 seconds

The order in which the commands are obeyed is as follows:

Address	Error print	Address
1231		0643
0650-0651		0467
0475		0654-0657

1021		1044
1245-1250		1570
0674-0710	0445	0614
0711		1601
0435-0441	0447	1425
0442		1451
1327		1675
0414		1321-1323
0501-0504		1347
0571-0601		1534
0517-0523	0605, 1463,	1060
	1361	
0524-0530	0612, 1532,	0545
	0507	
0531-0534	0617	1332
0535		0417
1561-1565		1604
0552-0556	1454	1430
0557		1515
1544-1550		1541
0631-0641	1421	1465
0741-0746		1052
1254-1256	1662	1076
1501-1505		1522
0731-0740		0646
1271-1277		1672
0422		1416
0607		1542
1574		1266
1261		0512
1242		0377
1523		0425
1010-1011		0406
1472		1773
1016		0564
1042		0751
1366		0426 jumps
		to 1232(end)
		second time
		through
1112-1113		0427-0430
0400		0651
1224		
0403		
1527		
1553-1556		

Part III

Part III contains two memory sum routines, which begin at 1310 and end at 1755. The first routine sums every cell in the main memory without regard to overflow, and prints out the sum in address 1774. The second routine sums every cell in the main memory after first shifting the contents fifteen binary places to the right to include the command portion in the sum. It prints out the total in address 1776. There is no check on these routines except to verify that the same sums are printed out on each run.

Because the entire main memory is summed, any command which is changed or added will alter the memory sum. If it is necessary to type-in commands for checking, it is best to operate in the buffer, since the latter is not included in the memory sum. However, addresses 1132-1177 are not used in the reliability routine.

Time: about 2 minutes.

Correct Sums* :

1774 35033611044332

1776 00365177127403

Remarks: Channel 16 contains a closed subroutine which is used to print the errors of the test commands of Part II. This routine operates with cell 3300 to obtain the address of the command which failed. It extracts the addresses of the operands of this command, and then prints the command and operands with addresses. One entry to this subroutine, for test commands which jumped when they should not have, is 1616. The other entry, for command that failed to jump, is 1621. The exit is 1641.

* First time through the routine, sums will be 1774 35174211044332
1776 00366121724023

Error Print-Outs: TABLES VI-1 and VI-2 show the possible types of error print-outs, with the location of the instructions in which the error occurred, in all three parts of the reliability program.

Procedure for Filling Reliability Routine from Punched Paper Tape

1. The tape for the routine contains carriage returns; therefore, to avoid double spaces, four tabs and the carriage return should be set up on the Flexowriter about 20 to 24 spaces apart, or the automatic carriage return should be removed.

2. Place the tape in the reader and press the START READ button. A routine will read in and clear the memory. Press the CLEAR button before filling the main routine.

3. Fill the routine by pressing the START READ button. The tape will fill channels 00 through 03 and stop. This is an independent section of the reliability routine and may be used without filling the rest of the computer.

4. To fill the rest of the routine, press the START READ button. Filling should take about 32 minutes.

5. To start computing with the entire routine in the memory, begin in 0000. Test Switches 2010 and 2100 should be ON, the other two Test Switches OFF.

6. The computer should type out the following for each run of the test:

The alphabetic mode uses every Flexowriter Key.

(Judge how quickly it "duz" it!)

```
0626 35210021002100
2100 00000000000000
2100 00000000000000
1774 35033611044332
2100 00000000000000
1776 00365177127403
```

NOTE

The first time through the routine the memory sums will be different. The second and succeeding times, the sums will be as shown above.

Procedure for Writing Reliability Routine on Magnetic Tape

1. Clear the memory and fill the reliability routine into the memory.

2. Fill the following program into the memory:

NOTE

Addition of the following program will change normal memory sums.

```
1200 14 f f 0003 bs for first
                                block
1201 35 1214 2100 1202 prestore block
1202 05 2100 2100 (f) bl
1203 15 f 0001 f wt, section 1
1204 30 1210 1211 1210 wait
1205 35 1202 1212 1202 augment bl
1206 34 1213 1202 1202 through?
1207 11 2100 2100 2000 yes, fill
1210                                0
1211                                200
1212                                10
1213 2100 2100 2000 limiter
1214 05 2100 2100 f dummy
(this routine writes the reliability rou-
tine on magnetic tape.)
1132 30 1140 1143 1140 wait
1133 16 f 0001 f rt
1134 34 1135 1141 2000 through?
1135 04 2100 2100(1700)no, fill
                                channel 17
1136 35 1135 1142 1135 augment the
                                bo
1137 34 3000 2100 1132 return
1140                                0
1141 2100 2100 1774 limiter
1142                                10
1143                                200
(this routine is part of the bootstrap rou-
tine to fill the reliability routine.)
```

3. Write Blocks 0, 1, and 2 onto

ERROR PRINTOUTS FOR PART I				
Printout	Commands Tested	Location of Instructions		Operation
		Start	End	
00000000000001 OXXXXXXXXXXXXX OXXXXXXXXXXXXX	ad tm	0001	0010	$\sum_{i=1}^{1023} i, i=1,2,3,4,5,\dots$
00000000000002 OXXXXXXXXXXXXX OXXXXXXXXXXXXX	ad tm md	0011	0032	$\sum_{i=1}^{511} i^2, \quad "$
00000000000003 OXXXXXXXXXXXXX OXXXXXXXXXXXXX	ad tm md	0033	0044	$\sum_{i=1}^{383} i^3, \quad "$
00000000000004 OXXXXXXXXXXXXX	su sm tm	0045	0051	$(1-2^{-36}) - \sum_{i=1}^{36} 2^{-i}$
00000000000005 OXXXXXXXXXXXXX	su sm tm	0101	0107	$(-1+2^{-36}) + \sum_{i=1}^{36} 2^{-i}$
00000000000006 OXXXXXXXXXXXXX	su sl tm	0110	0117	$(-1+2^{-36}) - \sum_{i=1}^{36} 2^{-i}$
00000000000007 OXXXXXXXXXXXXX	su sm to tm	0120	0130	$(1-2^{-36}) + \sum_{i=1}^{36} 2^{-i}$
00000000000010 OXXXXXXXXXXXXX OXXXXXXXXXXXXX	sf sm sl tm	0146	0155	$\exp(1-2^{-36})^p = p$
00000000000011 OXXXXXXXXXXXXX OXXXXXXXXXXXXX	sf	0166	0170	$\exp(0) = -100$
00000000000012 OXXXXXXXXXXXXX OXXXXXXXXXXXXX	mr ad su tm	0177	0210	$\sum_{i=1}^{16} ar^i$
error printout	ad(command)	0211	0214	
00000000000013 OXXXXXXXXXXXXX	dr ad tm	0215	0253	$a/b = q$
00000000000014 OXXXXXXXXXXXXX OXXXXXXXXXXXXX	dd md ad to	0254	0272	$a/b = q + r/b; a = bq + r$
Failure to print the following two lines: The alphabetic mode used every Flexowriter Key (Judge how quickly it "duz" it!)	ex pr 0300	0300	0351	

TABLE VI-1

ERROR PRINTOUTS FOR PARTS II AND III				
PART II Minimum access coding error: Four printouts with address, 1) address and command that failed. 2) address and m_1 of command that failed. 3) address and m_2 of command that failed. 4) address and m_3 of command that failed.	all previous commands	0400	1677	
		start in	1231	
PART III Memory sums are printed with address. They are in 1744 and 1766 and will always print out, and should be the same from run to run.		1310	1755	

TABLE VI-2

magnetic tape, using some empty memory cells for the write commands:

```

1700 04 2100 2100 1700 bo, obeyed in
                        the buffer
1701 04 2100 2100 1740 bo, obeyed in
                        the buffer
1702 34 3000 2100 1703 transfer,
                        obeyed in the
                        buffer
1703 16 f 0001 f rt
1704 04 2100 2100 1710 bo
1705 30 1751 1755 1751 wait
1706 16 f 0001 f rt
1707 04 2100 2100 1750 bo
1710 30 1751 1755 1751 wait

```

The following 2 blocks (1711-1757) are combined for the drum numbering.

```

1711 16 f 0001 f rt
1712 04 2100 2100 f bo, fill the
                        memory with
                        reliability
1713 35 1712 1716 1712 augment the
                        buffer out
1714 34 1752 1712 1710 through?
1715 34 3000 2100 1132 yes, go to fill
                        17th channel
1716                                10
1717                                0
1750                                0
1751                                0
1752 2100 2100 1700 limiter
1753                                0
1754                                0
1755                                200
1756                                0
1757                                0

```

4. Start computing in cell 1200. This will cause the contents of the 102-A's memory to be written into blocks 3 through 202 on the tape. After writing block 202, the Flexowriter paper tape reader will start automatically.

5. The paper tape (previously prepared) will fill the buffer with the following coding. The buffer contents are then

written into block 203 on the tape, using some empty 102-A memory cells for the necessary commands:

```

2000 35 (f) 2005 2005 add, for mem-
                        ory sum
2001 37 2006 2100 2006 to, and trans-
                        fer
2002 21 2005 2003 0001 print the sum
                        with address
2003 22 f f f halt
2004 1 f f
2005 0 memory sum
2006 35 2000 2004 2000 augment the
                        add command
2007 34 2000 2006 2002 through?

```

(memory sum sub-routine)

Procedure for Filling Reliability Routine from Magnetic Tape

1. Type the following program:

```

1776 1776 - 14 f f f tab
16 f 0001 f tab
compute

```

2. Routine will now fill reliability routine and then take a memory sum and print it out.

COMMAND TEST PROGRAM

The command test program contains a series of tests designed to use each of the arithmetic and logical commands. It is especially useful in marginal checking, as the maintenance engineer can vary a voltage and detect a failure by observing the behavior of the program counter, without waiting for a printout. Commands which fail may be repeated, using the test switches, or the program can be allowed to step automatically from command to command. Coding for the command test program is given in APPENDIX E.

One complete cycle of the program requires only about 20 seconds, permit-

ting the engineer to quick-check operation of the computer at selected limit values of marginal parameters.

After any failure which might have turned on a record flip-flop for one or more word times, it is recommended that the memory channels be cleared with the following program, to remove any P_2O_{13} pulses:

```

2000 2000 - 26 2100 2100 f
      2001 - 35 2000 2003 2000
      2002 - 34 3000 2100 2000
      2003                1
          S

```

The first portion of the program is a print test routine. Correct printouts:

```

0203 00123456765432
0204 02112233445566 001234567
      021122334
0205 +123456789
0206 -112233445 pl23456 n678901
National Cash Register

```

Test Switch 2020 can be used to repeat portions of the print routines. Failure of a print routine is determined by inspection. Test Switch 2010 UP will skip the print test routine.

In subsequent portions of the program, failure of a command is indicated by a jump to a Test Switch 2040 "stick" command. This will be indicated by a quasi-static condition of the neonlights on the test panel. A_1 will be on, N_1 , N_2 , N_3 will be on, N_4 , N_5 , N_6 , and N_9 will be dimly lit. When this condition occurs, Switch 2040 can be thrown DOWN-UP to give a diagnostic printout. If Switch 2100 is UP when 2040 is thrown, the test routine for the command which failed will be repeated. With 2100 DOWN, the program will continue to the next test in order after the printout.

If a command was induced to fail through intentional variation of a voltage (mar-

ginal check), that voltage should be returned to its correct value before throwing Switch 2040, to permit proper operation of the print subroutines.

Diagnostic printouts are as follows:

Command	Printout (all prints with address)
to	Command which failed, number tested.
tm, ta	Command which failed, numbers compared.
ad, su sl, sm	Command which failed, (m_1) and (m_2) of command,
ex	(m_3) of command result, correct result from storage.
sf	Command which failed. old exponent operand new exponent operand (scale factored) correct new exponent correct scale factored operand
md	Command which failed (m_1) and (m_2) of command. LSH of computed product MSH of computed product LSH of correct product MSH of correct product
mr	Command which failed (m_1) and (m_2) of command. rounded product (computed) unrounded product (previously computed) correct rounded product correct unrounded product
dd	Command which failed (m_1) and (m_2) of command.

	remainder (computed)	Command	
	quotient (computed)	Addresses	<u>Function</u>
	correct remainder	0067-73	Shift logical left
	correct quotient	0074-100	Shift logical right
dr	Command which failed	0101-05	Shift magnitude left
	(m_1) and (m_2) of command.	0106-12	Shift magnitude right
	rounded quotient (computed)	0113-17	Extract test
	unrounded quotient (previously computed)	0120-22	Scale factor
	correct rounded quotient	0123-25	Scale factor
	correct unrounded quotient	0126-30	Multiply double
		0131-33	Multiply double
		0134-36	Multiply double
		0137-41	Multiply double
		0142-44	Multiply rounded
		0145-47	Multiply double
		0150-52	Multiply rounded
		0153-55	Divide with remainder
		0156-60	Divide with remainder
		0161-63	Divide with remainder
		0164-66	Divide with remainder
		0167-71	Divide rounded
		0172-174	Divide with remainder
		0175-177	Divide rounded
		0200	TS 2020 repeats main routine from 0000
		0201	Halt

The following is a brief outline of the complete program:

<u>Command</u> <u>Addresses</u>	<u>Function</u>
0000	TS 2010 UP skips print routine (to 0007)
0001-05	Print octal, decimal, alphanumeric
0006	TS 2100 UP repeats 0001-05
0007	to (negative) jump to 0350 if error
0010	to (positive) jump to 0015
0011	TS 2040 error indication
0012-13	Error print (to)
0014	TS 2100 UP repeats 0010
0015	tm (negative) jump to 0355 if error
0016	tm (positive) jump to 0023
0017	TS 2040 error indication
0020-21	Error print (tm)
0022	TS 2100 UP repeats 0016
0023	ta (negative) jump to 0362 if error
0024	ta (positive) jump to 0031
0025	TS 2040 error indication
0026-27	Error print (ta)
0030	TS 2100 UP repeats 0024
0031-35	Add test
0036-42	Add test
0043-47	Add test
0050-54	Add test
0055-61	Subtract test
0062-66	Subtract test

Subroutine A

The main routine jumps to subroutine A if the computed result is not equal to the stored result.

<u>Command</u> <u>Addresses</u>	<u>Functions</u>
0300	TS 2040 error indication
0301-05	Construct exit links
0306-17	Set up print commands
0320-24	Error print commands
0325-26	Extract 0001 into m_3 of "result" print commands
0327	TS 2100 - optional exit - 2100 UP repeats test which failed.
0330	Unconditional exit to next test in main routine.

Subroutine C

Subroutine C contains the to, tm, and ta error printouts and the test routine for both halves of a double length result.

<u>Command Addresses</u>	<u>Function</u>
0350	TS 2040 error indication
0351-02	Error prints (to)
0353	TS 2100 UP repeats 0010
0354	Jump to 0011 (next test)
0355	TS 2040 error indication
0356-57	Error prints (tm)
0360	TS 2100 UP repeats 0016
0361	Jump to 0017 (next test)
0362	TS 2040 error indication
0363-64	Error prints (ta)
0365	TS 2100 UP repeats 0024
0366	Jump to 0025 (next test)
0367	0
0370-72	Construct exit link (double length test routine)
0373-0405	Set up error test commands
0406-0411	Error test commands (jumps to subroutine A if error)
0412-13	Restore error print commands to one word
0414	Unconditional exit to next test in main routine.

Procedure for Filling the Command Test Program (coding for all programs is APPENDIX E)

1. Clear the memory, using the program under "Erase Main Memory".
2. Fill the Command Test program.
3. Check the program for possible filling errors by running the Memory Sum and Shifted Memory Sum programs.
Memory Sum: 00246515226156
Shifted Sum: 00117302110415
4. Throw Test Switch 2010 DOWN, 2020 UP, 2100 DOWN.
5. Type two hyphens and depress the COMPUTE button, starting at 0000.

MARGINAL CHECKING

In order to determine the condition of many of the plug-in units and adjustments in the computer, the preceding programs may be run with certain computer voltages disturbed from their proper values by known amounts. The range over which such voltages may be varied without causing an operational error will be called the marginal range. TABLE VI-3 gives the voltages to be varied and the portions of the circuits affected by such variation.

<u>Voltage</u>	<u>Circuits Affected</u>
Noise Clipper Bias	Clamped and Unclamped Flip-Flops, Logical Nets
-300V Regulated	Drivers, Relay Drivers
Memory Clipping Levels	Memory Playback Circuits
Memory DC Levels	Memory Playback Circuits

TABLE VI-3.
Voltages Used in
Marginal Checking

The marginal range for each operational test should be recorded in the log each time a test program is run. Any significant deviation from normal marginal range is cause for investigation of the circuits affected by the marginal change. TABLE VI-4 gives acceptable marginal ranges for various modes of operation. These values are for independent (not combined) variation of the voltages. Complete instructions for varying the voltages is included under POWER SUPPLIES.

The procedure for running marginal checks is as follows:

- 1) With all voltages at their normal settings $\pm 1\%$, fill the program under investigation (see

list of programs in TABLE VI-4).

- 2) Reduce voltage to lower marginal limit, indicated in TABLE VI-4.
- 3) Run the program and note result. If there are no errors continue procedure. If an error occurs, check circuits affected and remove marginal component.
- 4) Raise voltage to the upper limit and run the program again. Note result as above.
- 5) If there are no errors, repeat, varying another voltage, or go to next program.

If desired, the margins can be varied gradually past the given limits, until error occurs, to observe the true marginal range for each program. The codings for all the programs mentioned in TABLE VI-4 are included in APPENDIX E.

MEASUREMENT AND ADJUSTMENT

POWER SUPPLIES

The measurement and adjustment points in the power supplies are located on four separate chassis. Figures VI-1 and VI-2 are photographs of Upper Power Supply No. 3 (UPS-3). Figures VI-3 and VI-4 are photographs of Upper Power Supply No. 4 (UPS-4). Figure VI-5 shows Lower Power Supply No. 2, and Figure VI-6 shows Lower Power Supply No. 4.

Voltages

The power supply voltages have been broadly separated into four groups, as follows:

1. A-C Voltages: 6.3v, 12.6v, 220v reg. (Stabiline).
2. Logical System D-C Voltages: +225, +100, noise clipper bias, grid clamp bias, -300.

Test	NCB	<u>M A R G I N A L R A N G E</u>	
		Memory Clipper	
		(Upper Level)	-300V
Any Filling	3.5 - 8.5	-	295-303
Any Printing	3.5 - 8.5	-	295-303
Command Test Program	2.5 - 8.5	-	295-303
Minimum Access Test*	2.5 - 8.5	18-30	295-303
Reliability Program Part I	3.5 - 8.5	-	295-303
Reliability Program Part II*	3.5 - 8.5	18-30	295-303
Tape Unit Read-Write Program	3.5 - 8.5	-	295-303
IBM Punch	3.5 - 8.5	-	295-303
IBM Read	3.5 - 8.5	-	295-303
* Should also run error-free with 4 volt difference between memory channel DC levels. (See "Memory DC Voltages" page , for level adjustment procedure.)			

TABLE VI-4. Marginal Range for Various Modes of Operation

3. Memory D-C Voltages: +425 memory record voltage, +250 reg., memory record driver clamp voltages (+150 and +115), and memory clipper bias (+10 and +20).
4. Miscellaneous D-C Voltages: +90 sequence relay supply, +45 reference battery -7 thyatron bias.

A-C Voltage Measurement: The 6.3- and 12.6-volt a-c filament supplies are not adjustable. Any unusual deviation from rated values presumably would indicate incorrect line voltage, defective transformer, defective tube (some filaments are connected in series), or a short in the filament wiring. An a-c voltmeter calibrated to $\pm 3\%$ or better should be used in checking filament voltages. The filament voltage checkpoints are listed in TABLE VI-5.

Measure Between	Correct Indication
Pins 7 and 8 of V1, UPS-3	6.3v ac $\pm 10\%$
Pins 3 and 4 of V7, UPS-4	6.3v ac $\pm 10\%$
Pins 3 and 4 of V10, UPS-4	6.3v ac $\pm 10\%$
Pins 7 and 8 of V9, UPS-4	6.3v ac $\pm 10\%$
Pin 3 of V2, Pin 4 of V4, UPS-3	12.6v ac $\pm 10\%$
Pin 7 of V5, Pin 8 of V6, UPS-4	12.6v ac $\pm 10\%$
TS1-61 and 62	12.6v ac $\pm 10\%$
TS2-64 and 65	12.6v ac $\pm 10\%$
TS2-61 and 62	12.6v ac $\pm 10\%$
TS2-64 and 65	12.6v ac $\pm 10\%$
TS3-5 and 6	12.6v ac $\pm 10\%$
TS3-7 and 8	12.6v ac $\pm 10\%$
TS3-61 and 62	12.6v ac $\pm 10\%$
TS3-64 and 65	12.6v ac $\pm 10\%$
TS4-61 and 62	12.6v ac $\pm 10\%$
TS4-64 and 65	12.6v ac $\pm 10\%$

TABLE VI-5 Filament Voltage Checkpoints

The 220-volt regulated ac (Stabiline) is adjusted to the correct value during the "+225-volt unregulated" adjustment procedure in the following paragraph.

Logical System D-C Voltages: As there is some interaction among the +225 unregulated, +125, +100, and -300 volt adjustments, the following adjustment procedure is recommended. These voltages should be measured with a d-c voltmeter calibrated to $\pm 1\%$.

(1) 225v (Unregulated):

- a) Measure between J1 and ground (see Figure VI-1) for 225v, or use Test Panel Voltmeter. DO NOT THROW SWITCH S1.
- b) Adjust 225 (unregulated) to 225v, using potentiometer under knurled cap on Stabiline (UPS-2, external). Start the adjustment with the potentiometer in the maximum counter clockwise position.
- c) Repeat step a, if the voltage cannot be brought to correct value with Stabiline adjustment, return Stabiline adjustment to minimum and change secondary tap on transformer, LPS-3 (see Figure VI-7).
- d) Make final adjustment to 225v supply by readjusting potentiometer on Stabiline.

(2) -300 (Regulated):

- a) Measure between ground and J10 (see Figure VI-3) for -300v or use Test Panel Voltmeter. DO NOT THROW SWITCH S4.
- b) Adjust -300 (regulated) to -300v, using R51.

c) Repeat Step a.

(3) +100v Regulator:

- a) Measure voltage between J5 and ground (see Figure VI-3)

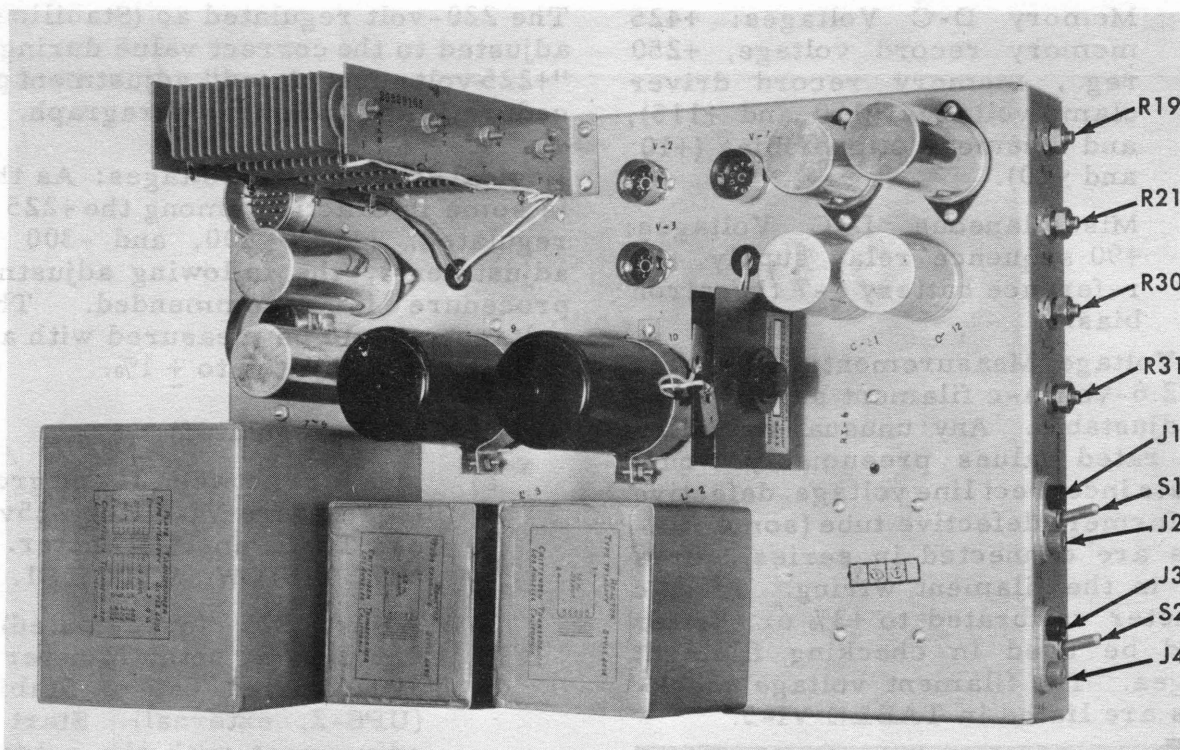


Figure VI-1. Upper Power Supply No. 3 — (internal view)

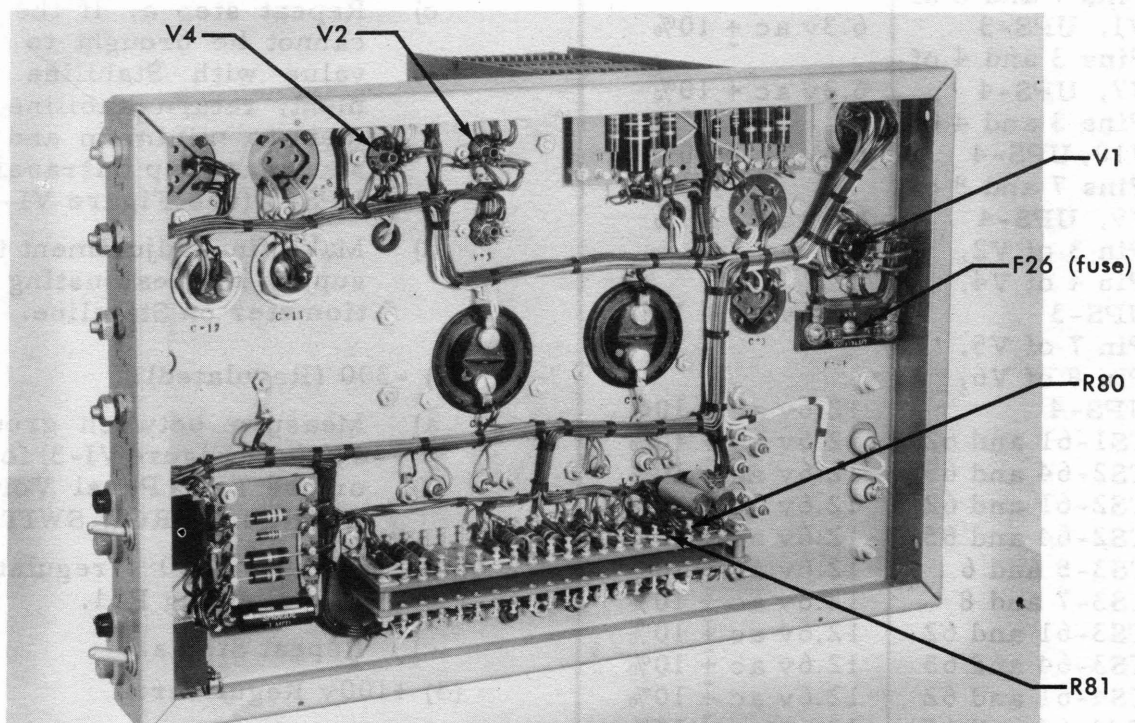


Figure VI-2. Upper Power Supply No. 3 (external view)

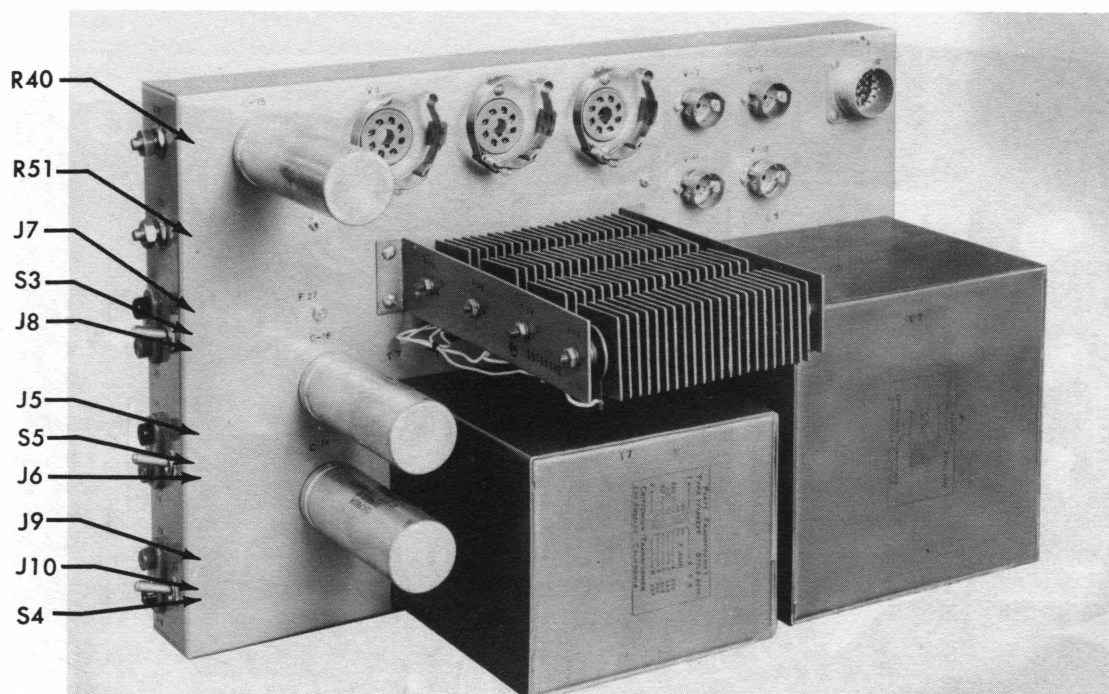


Figure VI-3. Upper Power Supply No. 4 (internal view)

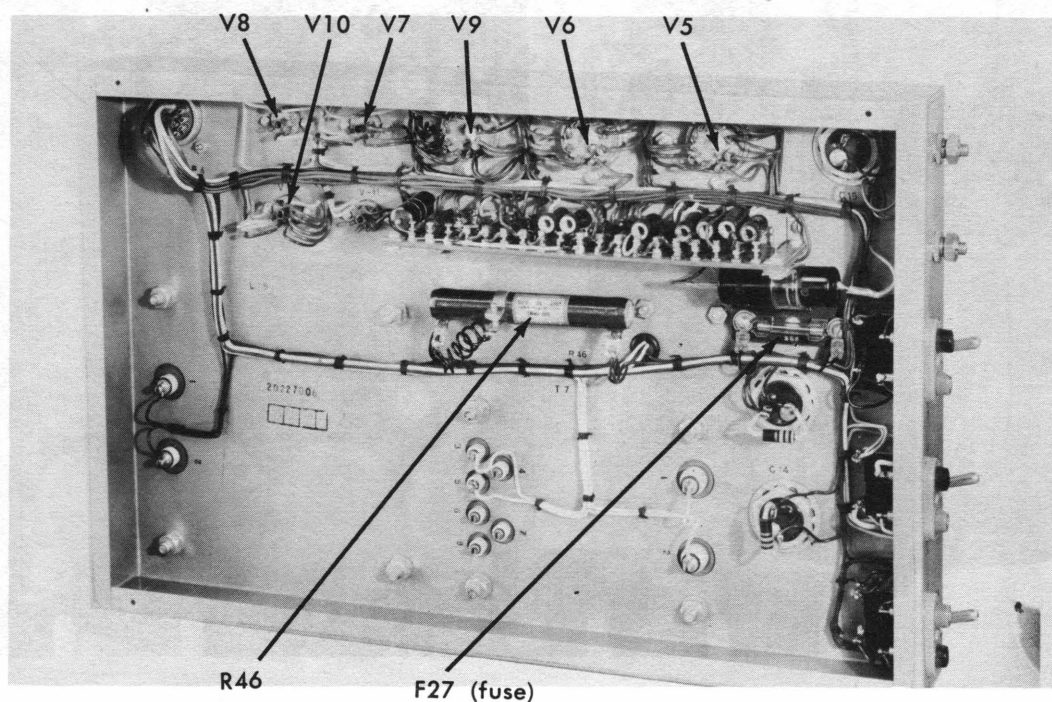


Figure VI-4. Upper Power Supply No. 4 (external view)

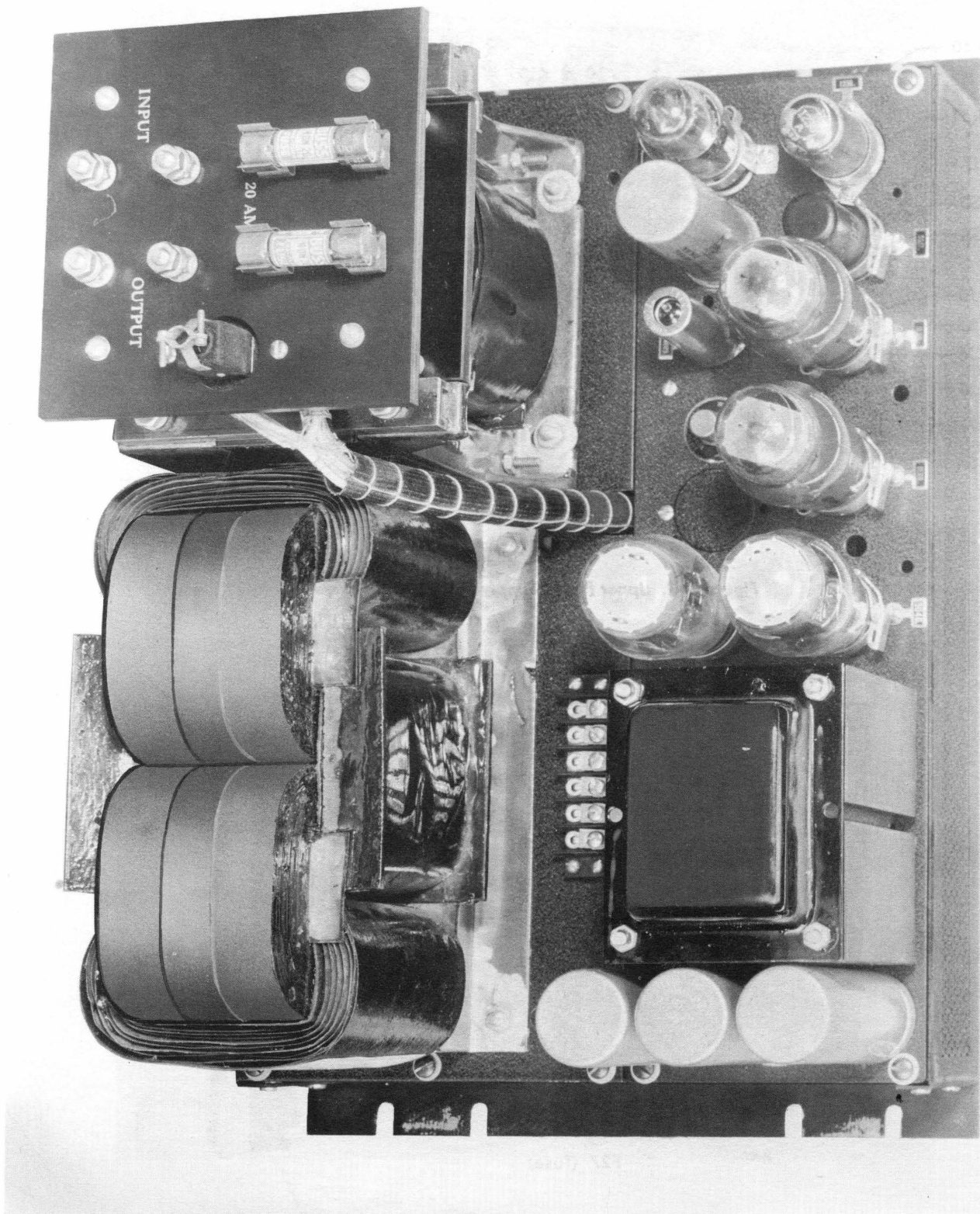


Figure VI-5. Lower Power Supply No. 2 (internal view)

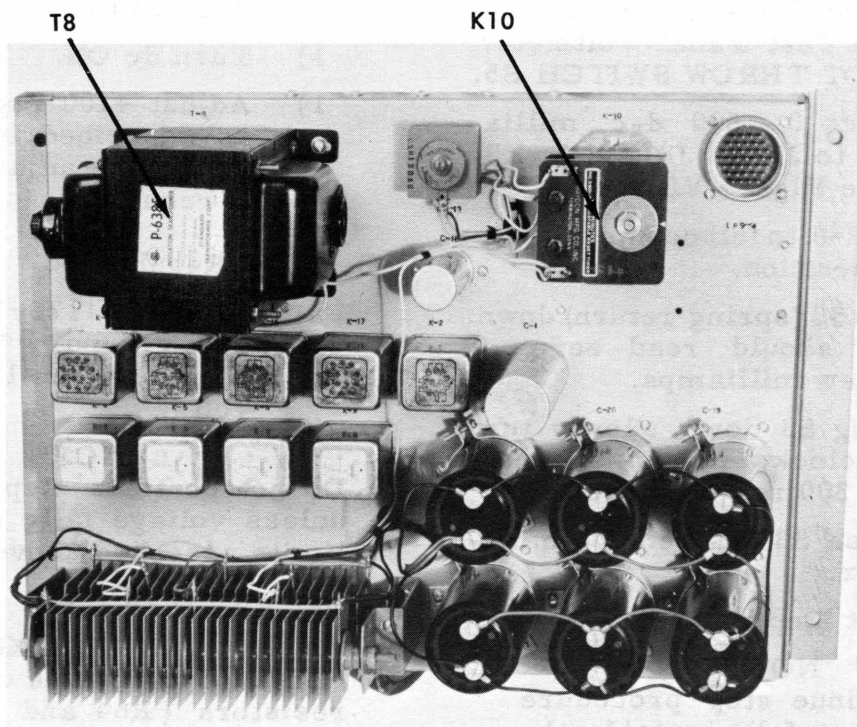
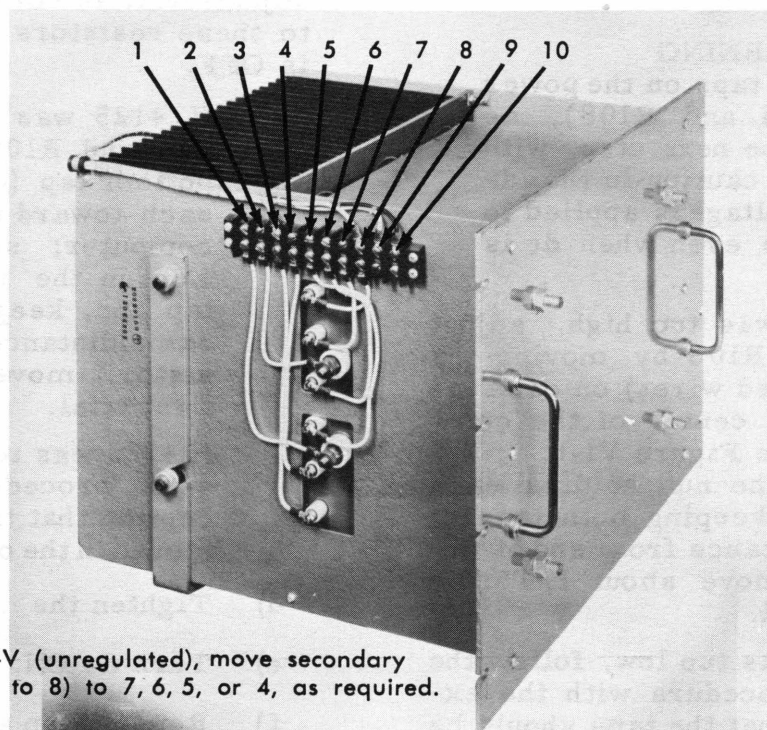


Figure VI-6. Lower Power Supply No. 4 (internal view)



NOTE:

To Adjust 225-V (unregulated), move secondary tap (connected to 8) to 7, 6, 5, or 4, as required.

Figure VI-7. Lower Power Supply No. 3 (external view)

for a reading of 99 to 101 volts, or use Test Panel Voltmeter. DO NOT THROW SWITCH S5.

- b) Connect 0 - 500 d-c milliammeter to J7 and J8 (red jack is +), see Figure VI-3.
- c) Turn R40 to full counter clockwise position.
- d) Throw S3 (spring return) down. Meter should read zero or very few milliamps.
- e) Holding S3 down, slowly turn R40 clockwise until meter reads 300 milliamperes.
- f) Release S3 and remove milliammeter.
- g) Repeat Step a.

NOTE

Do not continue step procedure unless voltage falls outside the limits specified above.

- h) Turn OFF dc.

WARNING

Never move the taps on the power resistors (R64 and R108), as mentioned in the next step, with the dc ON. Use caution in the adjustment, as voltage is applied to these resistors even when dc is OFF.

- i) If +100 was too high, adjust R64 and R108 by moving the top tap (red wires) on each toward the center of the computer; see Figure VI-8. Loosen the nut holding each tap and, keeping both taps at same distance from end of resistor, move about 1/4" for first trial.

If +100 was too low, follow the same procedure with the exception that the taps should be moved in the opposite direction.

- j) Tighten the nut on each tap.
- k) Turn dc ON.
- l) Adjust +100 regulator current as described in Steps a to g. Repeat the adjustment in Step i if required.

(4) +125v (Regulated):

- a) Measure +125v between TS2-10 and ground. Voltage should read between 123.5 and 126.5 volts.

NOTE

Do not continue step procedure unless voltage falls outside the limits specified above.

- b) Turn OFF dc.

WARNING

Never move the taps on the power resistors (R64 and R108), as mentioned in the next step, with the dc ON. Use caution in the adjustment, as voltage is applied to these resistors even when dc is OFF.

- c) If +125 was too high, adjust R64 and R108 by moving the bottom tap (orange wires) on each toward the outside of the computer; see Figure VI-8. Loosen the nut holding each tap and, keeping both taps at same distance from end of resistor, move about 1/4" for first trial.

If +125 was too low, follow the same procedure with the exception that the taps should be moved in the opposite direction.

- d) Tighten the nut on each tap.
- e) Turn dc ON.
- f) Repeat Step a. Repeat the adjustment in Step c if required.

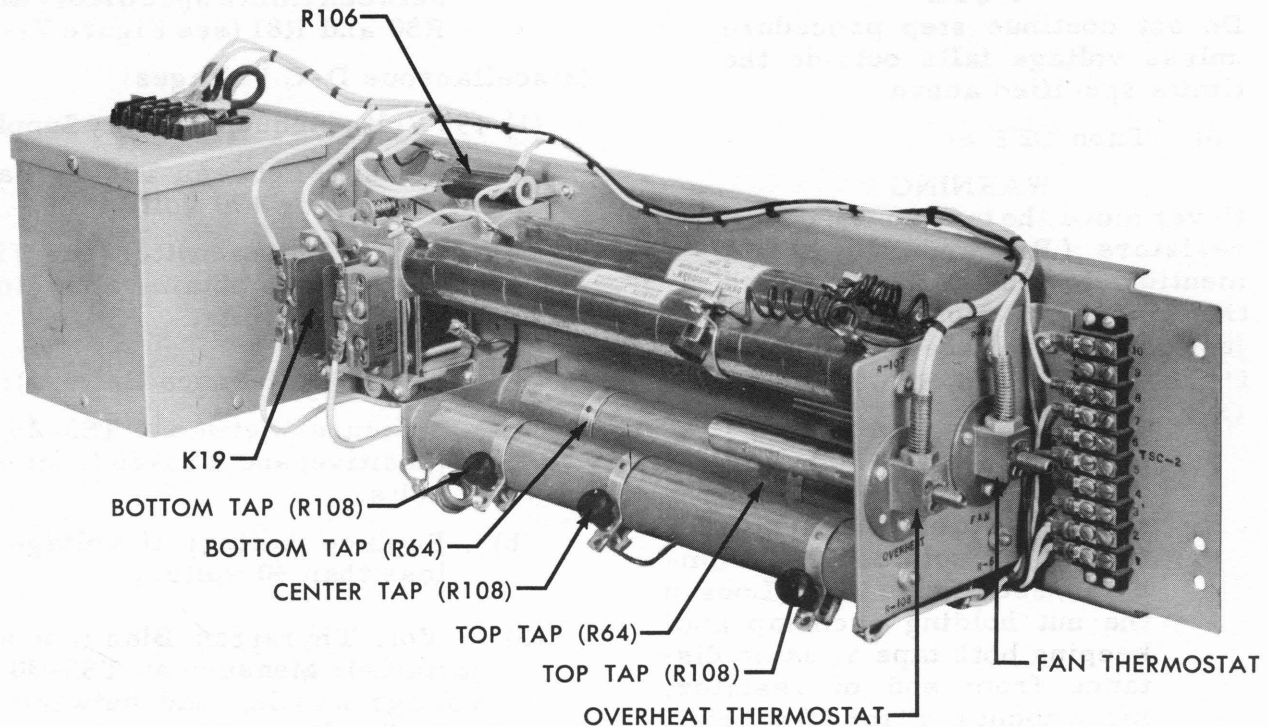


Figure VI-8. Fan Contactor Panel

- (5) Recheck -300v (J10 to ground, see Figure VI-3), +225v (J1 to ground, see Figure VI-1), +125v (TS2-10 to ground), +100v (J5 to ground, see Figure VI-3). Make any final adjustments as necessary.

(6) Noise Clipper Bias

- a) Measure for +6 volts between TS4-1 and ground, or use Test Panel Voltmeter.
- b) Adjust noise clipper bias to +6 volts, using R21 (see Figure VI-1).

(7) Grid Clamp Bias (LGC)

- a) Check for -8 volts between TS4-4 and ground, or use Test Panel Voltmeter.
- b) Adjust R31 (see Figure VI-1).

If desired, the Test Panel Voltmeter can be calibrated at the time the logical system d-c voltages are adjusted by switch-

ing the meter to the voltage being adjusted, and recording its reading (as percentage high or low) at the time the voltage is brought to the correct value. The calibration chart can be kept in the computer log or on a label on the test panel.

Memory D-C Voltages:

- (1) Record voltage (not adjustable): Measure between TS3-3 and ground for +425v dc (+10%, -5%).
- (2) +250v (Regulated):
 - a) Measure between J3 and ground (see Figure VI-1) for 250v. DO NOT THROW SWITCH S2.
 - b) Adjust R19 as required, starting adjustment with R19 in the maximum counter clockwise position.
- (3) +150v:
 - a) Measure between TS2-6 and ground for +150v (± 3 volts).

NOTE

Do not continue step procedure unless voltage falls outside the limits specified above.

- b) Turn OFF dc.

WARNING

Never move the taps on the power resistors (R64 and R108), as mentioned in the next step, with the dc ON. Use caution in the adjustment, as voltage is applied to these resistors even when dc is OFF.

- c) If +150 was too high, adjust R64 and R108 by moving the middle tap (blue wires) on each toward the outside of the computer; see Figure VI-8. Loosen the nut holding each tap and, keeping both taps at same distance from end of resistor, move about 1/4" for first trial.

If +150 was too low, follow same procedure with the exception that the taps should be moved in the opposite direction.

- d) Tighten the nut on each tap.
 e) Turn dc ON.
 f) Repeat Step a. Repeat the adjustment in Step c if required.
- (4) +115 (not adjustable): Measure between TS1-58 O and ground. Voltage should read between +112 and +116 volts.
- (5) Memory Clipper Bias (+10 and +20 volts):
- a) Measure voltage at TS3-60 O or TC4-C3-15; voltage should range between +11 and +13 volts.
- b) Adjust R30 (see Figure VI-1) to +24 volts. Check voltage at TS3-59 O or TC4-C3-13.
- c) Repeat Step a. If voltage is not

between limits specified, check R80 and R81 (see Figure VI-2).

Miscellaneous D-C Voltages:

- (1) +90 Volts (Sequence Relay Supply):

- a) Measure between TS4-37 I and ground for +90 volts.
- b) Adjust T8 tap switch (see Figure VI-6) to obtain value closest to +90 volts.

- (2) +45 volts (Reference Battery):

- a) Measure between TS3-25 O (positive) and TS3-26 O for +45 volts.
- b) Replace battery if voltage is less than 40 volts.

- (3) -7 Volt Thyatron Bias (not adjustable): Measure at TS3-30 I; voltage should read between -6 and -8 volts.

Ripple and Adjustment Range

Ripple and noise should be checked with a calibrated a-c oscilloscope having sufficient sensitivity to show one volt peak-to-peak signals. Adjustable supplies should be capable of variations over the ranges shown under Minimum Adjustment Range in TABLE VI-6.

Maximum figures given are for ripple at 60 or 120 cycles plus superimposed random noise, not including long-time (1/2 sec) drifts due to load changes or spikes, such as from line transients.

MEMORY**Head Installation and Radial Adjustment**

The following head installation and radial adjustment instructions apply to all heads in the 102-A computer.

- 1) Turn computer ON and run for at least 20 minutes to bring system up to

Supply	Measurement Point	Max. P-P Ripple and Noise	Minimum Adjustment Range
+425	TS3-3	2v	R19 (Figure VI-1) at least 15v (Potentiometer LPS-2) 215-235v
+250	J3 (See Figure VI-1)	1v	
+225	TS2-5 or J1 (Figure VI-1)	6v	
+200	Memory Clamp Bracket	1v	
+190	Memory Clamp Bracket	1v	
+150	TS2-6	1v	
+225	TS2-10	1v	
+115	TS1-58 0	1v	
+100	J5 (Figure VI-3)	1v	
+90	TS3-3	(Not Filtered)	
+24	TS3-59 0	0.3v	Measure at J7-J8 vary R40 (Figure VI-3) 0-450 ma*
+12	TS3-60 0	0.3v	
+6	Center Terminal R21 or RS4-1	0.4v	
-8	Center Terminal R31 or TS4-4	0.3v	
-7	TS3-30I	0.4v	
-300	J10 (Figure VI-3)	1v	
* WARNING: Do not exceed 500 ma in checking this adjustment. Throw switch S3 only while meter is connected in circuit.			

TABLE VI-6. Adjustable Voltages

normal operating temperature. Turn drum OFF. (see Figure I-3, page for location of drum motor ON-OFF switch.

CAUTION

To prevent possible damage to drum surface, make certain that drum is stopped before installing head.

2) Thoroughly clean area around the drum sleeve, making sure there is no dust, paint, or other foreign matter in bare and mating shoulders.

3) Release brass set screw so that plunger is free in mount; do not loosen brass screw more than is required to produce freedom of plunger. Head must slide freely in head mount (against spring ten-

sion) without scraping or binding.

NOTE

A new type of head mount construction, which is used on some of the 102-A computers, utilizes a hex socket #8-32 stainless steel set screw in place of the aforementioned brass set screw. This screw must be removed in order to install the head mount assembly in barrel. All other adjustments are made in the same manner described for the brass set screw.

4) Turn radial adjustment ringnut several turns clockwise so that head will not hit the drum when mount is inserted in sleeve. Set tangential adjustment (if any) to midpoint of travel, with tangential adjusting tool. (Figure VI-9 shows the head adjustments set and a magnetic head).

5) Insert headmount in sleeve and secure with headmount retaining screw.

6) Turn ringnut counterclockwise with ringnut tool until head just touches drum, then turn slightly clockwise.

7) Insert threaded plug in outer end of headmount. Attach clamp and dial gauge assembly. Insert ringnut tool. Adjust dial gauge position such that pointer is near center of scale when feeler rests on threaded plug (see Figure VI-10).

8) Turn ringnut counterclockwise until dial gauge stops moving; note reading. Make sure that this is actual contact point with drum by backing head off, running it in again, and observing that the readings agree.

9) Turn ringnut clockwise until dial reads either .001" (for radial-adjust-only heads) or .002" (for tangential adjust heads) greater than the contact point reading. Watch gauge as setscrew is tightened. If gauge reading changes as setscrew is tightened, loosen set-

screw slightly and compensate for the change by turning the ringnut. When setscrew is tight, readings should be as given above.

10) Remove gauge assembly, ringnut tool, and plug.

11) Turn drum by hand as final check, listening for any scraping sound.

Tangential Adjustment

The following instructions apply to all head mounts with tangential adjustments. Installation and radial adjustment for the tangential adjust heads are given under the preceding sub-head.

1) Turn drum ON and, with the eccentric adjust tool, move head in either direction until head is brought into correct pulse position. Determination of correct pulse position is given under a following sub-head titled "Adjustment of Recirculating Registers".

NOTE

Since the head is .002" from the drum, it may be required to set the preamplifier gain at maximum in order to compensate for amplitude loss.

2) Turn drum OFF and, after drum has stopped, readjust head radially to a .001" clearance. Radial adjustment procedure is given under the preceding sub-head.

3) If head position does not fully agree with proper pulse position, slight tangential adjustments can be made to compensate for this error.

Clock Channel

Observe clock waveform (C) at TC1-A9-5 or 1(31R)K. Adjust potentiometer at MTC 3/4 B (clock preamplifier) until the clock uptime is between 50 and 60 percent of the clock period. For this condition the preamplifier output at MTC 3/4 B pin 8 will be between 40 and 60 volts, peak-to-peak. The clock fall time at TC1-A9-5 or 1(31R)K should be

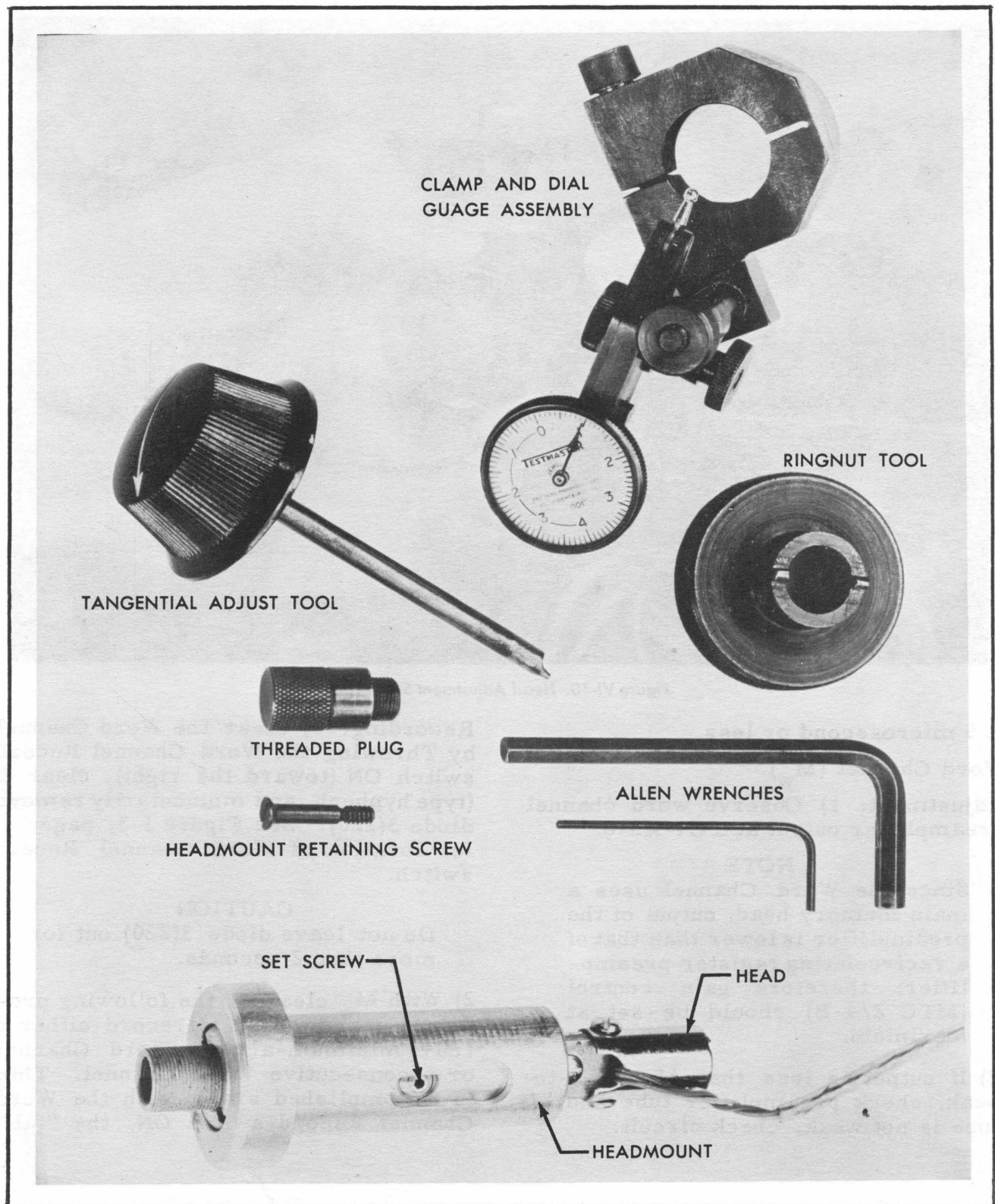


Figure VI-9. Head Adjustment Set and Headmount

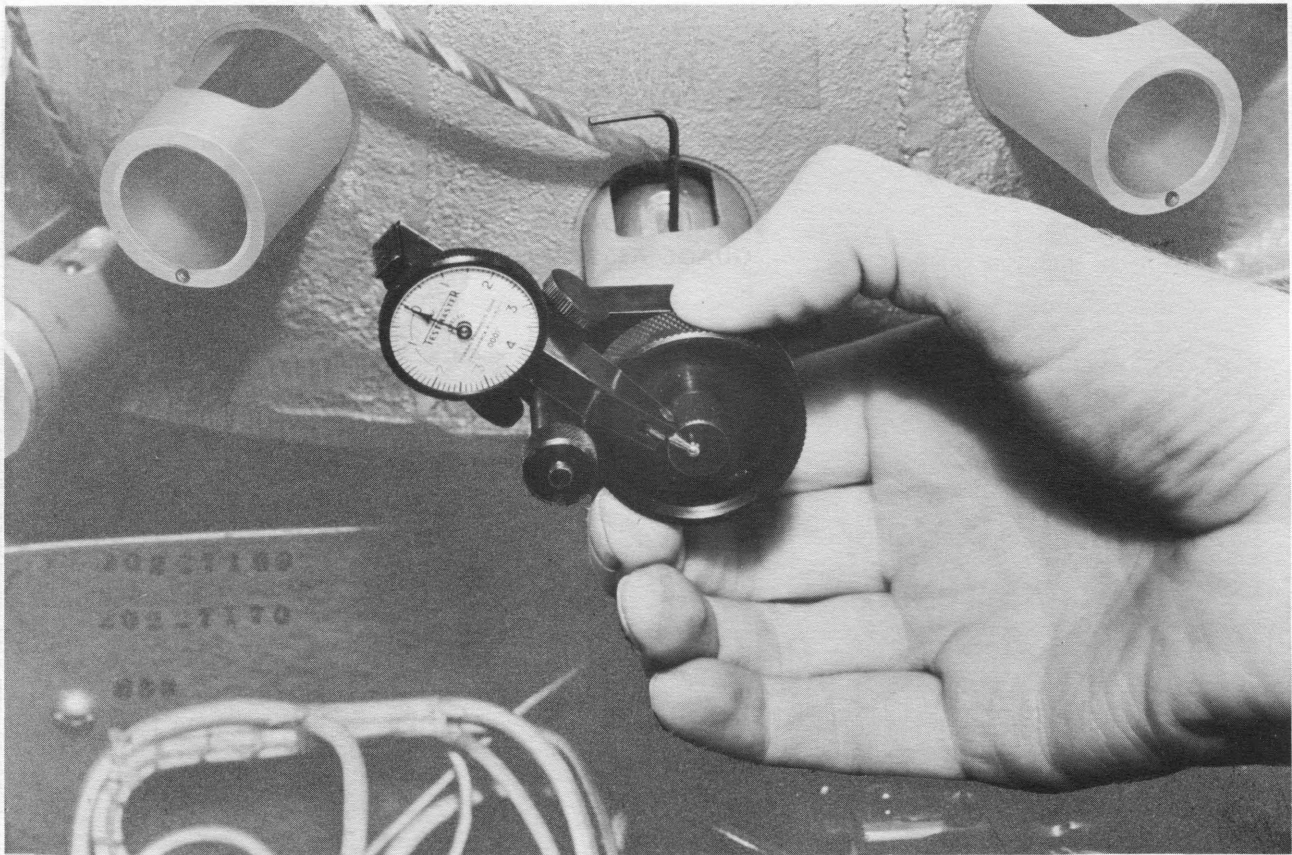


Figure VI-10. Head Adjustment Set In Use

0.5 microsecond or less.

Word Channel (M_w)

Adjustment: 1) Observe word channel preamplifier output at TC4-A5-8.

NOTE

Since the Word Channel uses a main memory head, output of the preamplifier is lower than that of a recirculating register preamplifier; therefore gain control (MTC 2/4-B) should be set at maximum.

2) If output is less than 65v peak-to-peak, check preamplifier tube and, if tube is not weak, check circuit.

Recording: 1) Clear the Word Channel by Throwing the Word Channel Record switch ON (toward the right), clear E (type hyphen), and momentarily remove diode 3(220). See Figure I-3, page for location of Word Channel Record switch.

CAUTION

Do not leave diode 3(220) out for more than 2 seconds.

2) With M_w cleared, the following programs may be used to record either a 180° minimum-access Word Channel or a consecutive Word Channel. This is accomplished since, with the Word Channel Record switch ON, the "tab"

routine transfers the contents of E to M^w instead of M in program counter block No. 47 (see Flow diagram). For example, if m₃ of H is 0000, the first tab will transfer the first sector address word from E to some random location on the drum. This address can then be looked up in order to record the next

sector address word. This process is continued around the channel.

NOTE

For proper operation of the synchronizing circuits, the sign portion of each sector address word must be 02, except sector 0 which must have 22 in the sign portion.

180° Minimum-access Word Channel Record Program:

-o0000-02004100410041	0041-02000200020002
0002-02004300430043	0043-02000400040004
0004-02004500450045	0045-02000600060006
0006-02004700470047	0047-02001000100010
0010-02005100510051	0051-02001200120012
0012-02005300530053	0053-02001400140014
0014-02005500550055	0055-02001600160016
0016-02005700570057	0057-02002000200020
0020-02006100610061	0061-02002200220022
0022-02006300630063	0063-02002400240024
0024-02006500650065	0065-02002600260026
0026-02006700670067	0067-02003000300030
0030-02007100710071	0071-02003200320032
0032-02007300730073	0073-02003400340034
0034-02007500750075	0075-02003600360036
0036-02007700770077	0077-02004000400040
0040-02000100010001	0001-02004200420042
0042-02000300030003	0003-02004400440044
0044-02000500050005	0005-02004600460046
0046-02000700070007	0007-02005000500050
0050-02001100110011	0011-02005200520052
0052-02001300130013	0013-02005400540054
0054-02001500150015	0015-02005600560056
0056-02001700170017	0017-02006000600060
0060-02002100210021	0021-02006200620062
0062-02002300230023	0023-02006400640064
0064-02002500250025	0025-02006600660066
0066-02002700270027	0027-02007000700070
0070-02003100310031	0031-02007200720072
0072-02003300330033	0033-02007400740074
0074-02003500350035	0035-02007600760076
0076-02003700370037	0037-22000000000000

Consecutive Word Channel Record Program

-o0000-02000100010001	0001-02000200020002
0002-02000300030003	0003-02000400040004
etc.	
0076-02007700770077	0077-22000000000000

Recirculating Registers

The following procedures cover the preamplifier output adjustments and the tangential head adjustments for the recirculating registers.

Gain Adjustment: 1) Observe the outputs of the E, F, G, H, and J preamplifiers with a calibrated a-c oscilloscope. Output waveform should have an amplitude of approximately 70 volts peak-to-peak. Preamplifier outputs and gain controls are as follows:

Register	Output	Gain Control
E	TC4-A1-8	MTC 2/4 A
F	TC1-C1-8	MTC 1/2 A
G	TC1-C1-9	MTC 3/4 A
H	TC1-C6-8	MTC 1/3 A
J	TC1-C6-9	MTC 1/2 B

In order to obtain a consistent output signal for measurement purposes, stuff a pulse into the register under observation by removing the following diodes:

Register	Remove Recirculation Diode	Remove Stuffing Diode	Stuffed Pulse
E	2(11J)	2(13J)	$P_0 O_{12}$
F	1(3B)	1(17B)	$P_2 O_{12}$
G	1(23G)	1(11G)	$P_0 O_1$
H	1(20N)	1(22N)	$P_1 O_0$
J	1(50N)	1(51N)	$(P_2 O_{13})^*$

2) Turn preamplifier gain to maximum and observe output amplitude. If amplitude is less than 75 volts, check preamplifier tube on g_m tester; replace if necessary. If tube is not weak, locate the cause of the low output and repair as needed. Repeat this step for all five registers.

3) Readjust each preamplifier output to 70 volts peak-to-peak.

NOTE

Preamplifier output should be nearly symmetrical about horizontal axis. Lack of symmetry usually indicates defective tube.

4) Replace all diodes that were removed.

Tangential Head Adjustment: 1) Stuff a pulse into register under observation by removing diodes indicated in step (1) of previous sub-head.

2) Blank scope with sampling clock signal (C_s). This signal is obtained at 4 (A51)K.

CAUTION

Use a .05 to .1 mfd capacitor in series with the blanking lead to the scope.

3) Trigger scope sweep with $P_2 O_{13}$ (fall) obtained at 3 (51DD)K.

4) Obtain preamplifier output of register being observed from point shown in step (1) of previous sub-head. Adjust sweep speed and horizontal position such that pulse occupies center of scope screen. Sweep should be from left to right. When observing the E or G line, adjust for 4 C_s blanking pulses between playback (large) pulse and record crossfeed (small) pulse; when observing the F or H line, adjust for 1 C_s blanking pulse between playback pulse and record crossfeed pulse. Figure VI-11 shows the correct waveforms for the E or G and F or H lines.

5) With correct tangential adjustment, blanking pulses should coincide with peaks of preamplifier output signal. To adjust, insert tangential adjustment tool in read head of recirculating register being adjusted and slowly change adjustment. Increase intensity to override blanking enough to make peaks of waveform visible on scope and watch preamplifier output signal for any increase in amplitude as the tangential adjustment is being made.

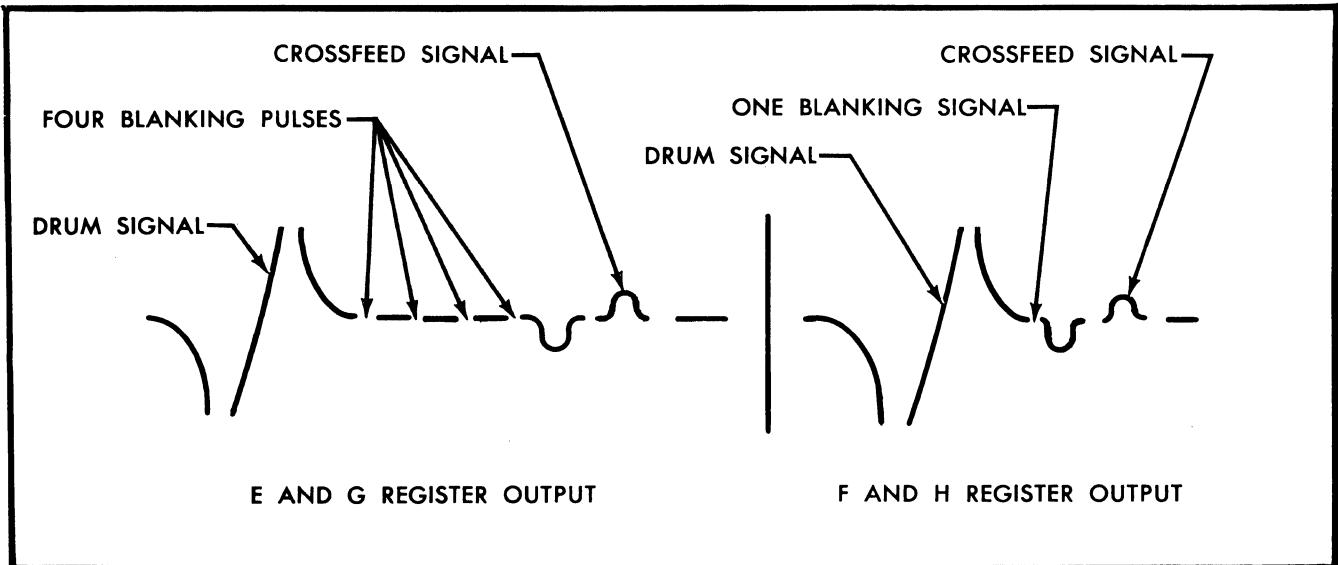


Figure VI-11. Output Waveforms for Recirculating Registers

CAUTION

If amplitude increases, do not continue adjustment as this indicates that the head is moving closer to the drum and may make contact. Instead estimate amount of adjustment required, turn OFF drum, turn ring nut counterclockwise, make estimated tangential adjustment, readjust radially as indicated under "Head Adjustment", restart drum, and continue as in step 5 previously.

6) When blanking pulses are nearly coincident with the peaks of the preamplifier output, readjust head radially to within .001" of drum and make fine tangential adjustment as follows:

a) Readjust preamplifier gain for a 70-volt peak-to-peak output.

b) Observe position of C_s blanking pulse at appropriate clipper outputs (see Figure VI-12 and also TABLE VI-7).

Register	Clipper Outputs	
	"ON"	"OFF"
E	4 (54A)K	4 (52A)K
F	1 (28A)K	1 (25A)K
G	1 (28D)K	1 (25D)K
H	1 (28I) K	1 (25I) K
J	1 (28K)K	1 (25K)K

c) Make tangential adjustment to center blanking pulse in clipper output waveforms, observing both clipper outputs after each change in adjustment. If it is impossible to center blanking pulse in both outputs, make best compromise adjustment.

7) Repeat tangential adjustment as needed on other registers.

8) Replace all diodes.

Main Memory

1) Adjust +250v regulated supply.

2) Synchronize oscilloscope at 4(43V)P. Connect oscilloscope input to limiter amplifier input (MTC-3/4-E-9 or stand-off insulator holding the common green output wire of the main memory plug-in units (MTC).

3) Type two hyphens. Type 25 2525 2525 2525 tab. This fills a one-zero pattern into Channel 0 Sector 00, which should be observable on the oscilloscope screen with sweep adjusted to approximately one word-time.

4) Note that potentiometers on memory plug-in units (20267339) are concentric. The inner (long) shaft is the gain control. The outer shaft is the d-c level control.

E REGISTER WAVEFORMS FOR TANGENTIAL HEAD ADJUSTMENT AND PREAMPLIFIER

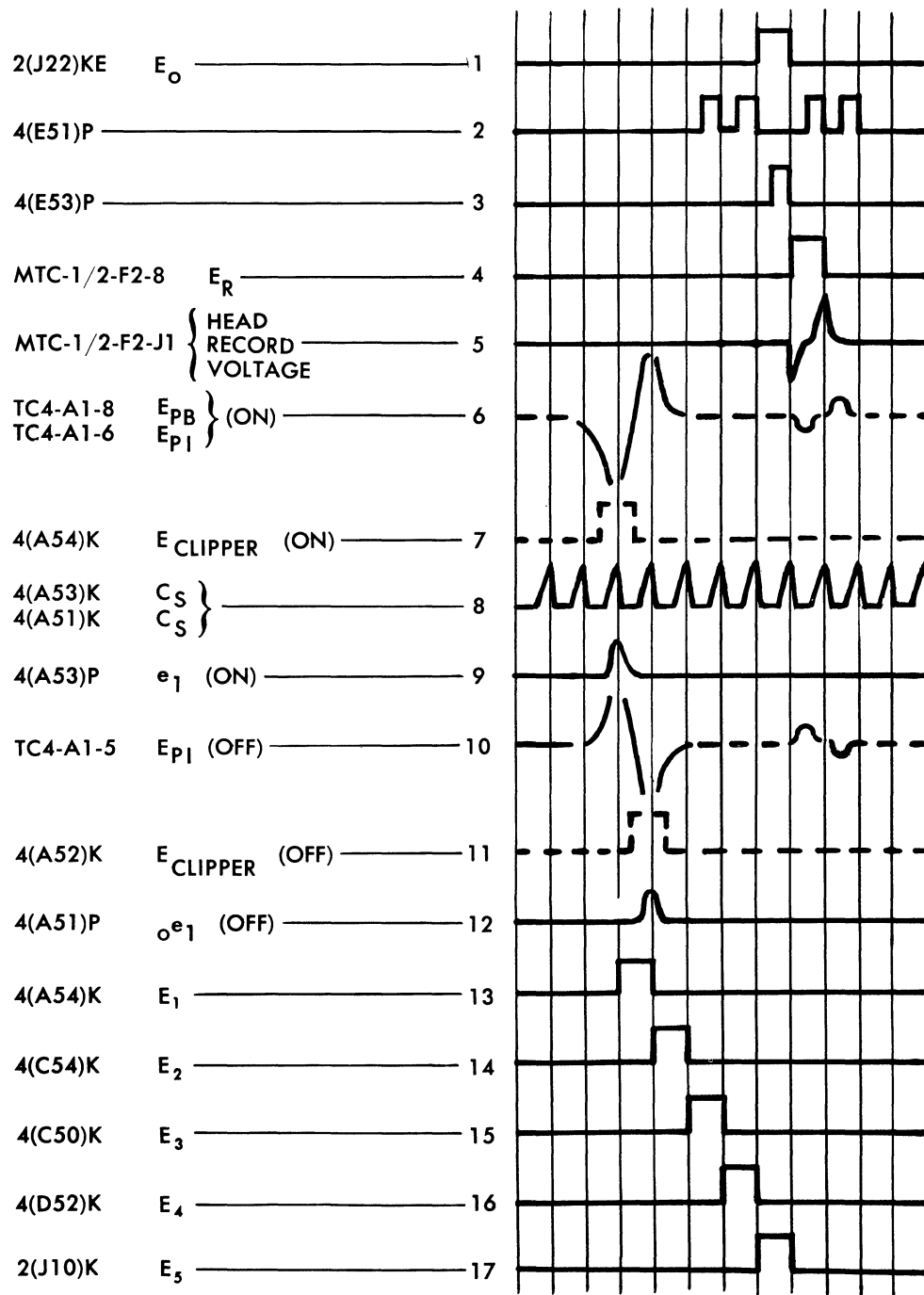


Figure VI-12. E Register Waveforms

Register	Check Record Pulse Steepness at	Preamplifier Voltage Output	Tangential Adj. for Time Between Drum Signal and Crossfeed Signal	Fine Tangential Head Adjustment Clipper Outputs	Check for Timing
E	E_r at MTC-1/2-F2-8	65-75v	4 C_s	4 (A54)K and 4 (A52)K	E_5 at 2 (10J)K coincide with 3_0 at 2 (22J)K
G	G_r at MTC-2/4-E2-8	65-75v	4 C_s	1 (D28)K and 1 (D25)K	G_5 at 1 (G22)K coincide with G_0 at 1 (17G)K
F	F_r at MTC-1/3-E2-8	65-75v	1 C_s	1 (A28)K and 1 (A25)K	F_2 at 1 (B2)K coincide with F_0 at 1 (14B)K
H	H_r at MTC-1/3-E2-8	65-75v	1 C_s	1 (I28)K and 1 (I25)K	H_2 at 1 (N19)K coincide with H_0 at 1 (24N)K
J	J_r at MTC-1/2-F2-8	65-75v	no crossfeed signal	1 (K28)K and 1 (K25)K	J_2 at 1 (50N)K coincide with J_0 at 1 (49N)K
NOTE: Sync scope with P_2O_{13} , sweep left to right. Blank scope with C_s at 4(A51)K through a 0.1 μ f capacitor.					

TABLE VI-7. Alignment Procedures for Recirculating Registers

Turn gain potentiometer on Channel 0 full ON (clockwise). Set d-c level potentiometer to halfway position.

5) Adjust main memory clamp levels (potentiometer, outer end of MTC 1/3) such that main memory playback is centered in clamping range. Position potentiometer halfway between points at which limiting of the playback waveform is observed.

Signal amplitude should be at least 3.5v peak-to-peak. If lower, locate cause of weak signal and correct. (Head output should be at least 0.9v peak-to-peak.)

6) Observe input signal to main memory phase inverter at TC3-B3-8 and adjust Channel 0 gain potentiometer for 70v peak-to-peak signal. If signal cannot be brought to 70v peak-to-peak amplitude, check main memory limiter

amplifier (20267340: MTC 3/4 - E) and correct trouble.

7) Return scope input to standoff insulator terminal on MTC (see step 2) and roughly adjust d-c levels of all channels as follows:

Switch scope to DC. Set channel selectors to 0 (type hyphen, hyphen, tab). With scope gain as high as possible, (vertical position control at lowest point) note level of Channel 0.

Without changing scope adjustments, set channel selectors to Channel 7. (Type 0700, hyphen, tab.) Adjust Channel 7 d-c level potentiometer (see TABLE VI-8 for location) to bring level to same value as observed for Channel 0. Repeat for Channels 1 through 17, remembering to change channel selectors each time. (For Channel n, type n00, hyphen, tab.)

8) Adjust gain for Channels 1 through 17, first filling 25 2525 2525 2525 in Sector 00 of channel (for Channel n, type n00, hyphen, 25 2525 2525 2525, tab), then turning gain potentiometer full ON (do not disturb d-c level potentiometer) and checking for 3.5v amplitude requirement, and finally setting preamplifier input signal to 70v peak-to-peak as observed at TC3-B3-8

NOTE

It is more convenient to adjust the gains and d-c levels in the order shown in TABLE VI-8 than in numerical order.

9) Make accurate adjustments of the d-c levels as follows:

Adjust scope sweep for one drum revolution by obtaining one synchronizing pulse from diode 4 (43V)P at each end of sweep. Trigger from PC 134 obtained at 3 (38F)P and adjust gain for approximately 4v (AC full-scale). Erase memory with

Program 7 in APPENDIX E.

Fill memory with test pattern by using Program 6 in APPENDIX E.

Fill and run Program 8 in APPENDIX E. This program starts by displaying output of Channel 0. When switch 2020 is thrown down-up, then switch 2100 is thrown down-up, the program compares output of Channel 1 to Channel 0. Channel 1 d-c level potentiometer can then be adjusted precisely. (do not change Channel 0 level potentiometer).

Throwing switch 2020 down-up, then switch 2100 down-up, compares Channel 2 to Channel 0, and the Channel 2 d-c level can be accurately adjusted.

Repeating the switch cycle advances the comparison one channel (in numerical sequence) permitting accurate d-c level adjustment of all channels against Channel 0. (Do not disturb gain potentiometer settings).

NOTE

It is discovered that one or more channels cannot be adjusted to match the d-c level of Channel 0, note the discrepancies and reset the Channel 0 d-c level to a value to which all channels can be adjusted, then repeat Program 8 in APPENDIX E from the beginning. If a contradiction occurs, (one channel will not go as high as the rest, another will not go as low as the rest) check tubes, then components in the 20267339 plug-in units involved, and correct the situation.

10) Final adjustments are made as follows:

With Program 8 in APPENDIX E running readjust clamp levels (potentiometer, outer end of MTC 1/3) such that record transient is clamped symmetrically above and below the base line.

Channel		Gain Control (Long Shaft)	DC Level Control (Outer Shaft)
0	MTC 1/2 - D	R14	R16
7	MTC 1/2 - D	R10	R11
6	MTC 1/2 - C	R14	R16
5	MTC 1/2 - C	R10	R11
4	MTC 1/3 - D	R14	R16
3	MTC 1/3 - D	R10	R11
2	MTC 1/3 - C	R14	R16
1	MTC 1/3 - C	R10	R11
10	MTC 2/4 - C	R10	R11
17	MTC 2/4 - C	R14	R16
11	MTC 2/4 - D	R10	R11
12	MTC 2/4 - D	R14	R16
13	MTC 3/4 - C	R10	R11
14	MTC 3/4 - C	R14	R16
15	MTC 3/4 - D	R10	R11
16	MTC 3/4 - D	R14	R16

TABLE VI-8. Main Memory Gain and DC Level Controls.

Observe output of M. M. limiter amplifier at TC3-B3-8 and adjust the limiter amplifier potentiometer (MTC 3/4 - E) for symmetrical clamping at this point. Refill Program 8 in APPENDIX E and trim gain potentiometers in all channels for 70v peak-to-peak signal at TC3-B3-8 (do not disturb d-c level potentiometers). The record transient observed at this point should not be more than 150v peak-to-peak.

2. Check tension of motor belt. Adjust tension as necessary to prevent slipping during carriage return.
3. Remove and clean rubber power roller with carbon tetrachloride or lacquer thinner.

Monthly:

1. Clean carriage return clutch with carbon tetrachloride or lacquer thinner.
2. Clean oil off reader contacts with nonabrasive paper.
3. Clean oil off carriage return and tab contacts with nonabrasive paper. (To reach this switch it will be necessary to disconnect the selector assembly and selenium rectifier.
4. Put one drop of oil on each end of the translator clutch sleeve, working the oil into the interior of the clutch.

FLEXOWRITER MAINTENANCE

The following schedule is noted on the Preventive Maintenance Check List Sheets. Refer also to Commercial Controls Corp. Service Manual (Parts II and VIII) on Model FL Flexowriter for descriptions and locations of components.

Biweekly:

1. Inspect drive chain and oil or grease when necessary.

Quarterly:

A complete overhaul is recom-

mended by the Commercial Controls Corp. This overhaul can be performed by Commercial Controls Corp. or by National Cash Register Company.

MISCELLANEOUS

RELAYS

WARNING

Disconnect computer's main power cable before checking Relays K18, K19, K20, and K21.

Manually operate contactors K18, K20, K21 (see Figure VI-13), and K19 (see Figure VI-8). Check armatures for binding or wear, and observe that springs are intact and spring tension is adequate. K19 and K20 should "snap" in closing and opening. Do not clean contacts unless such action is specifically indicated by a failure.

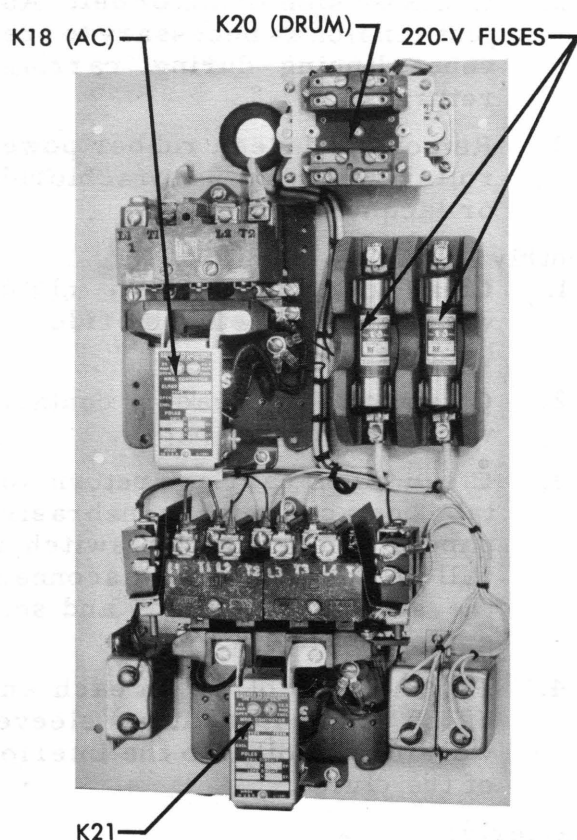


Figure VI-13. Contactor Panel

Check the time delay relay, K10 (see Figure VI-6), for approximately 50 seconds delay in the "power on" sequence. Adjust if necessary.

All relays should close without chatter in the power off-on and d-c off-on sequences. In replacing relays refer to the Relay Conversion Chart in APPENDIX F.

Sealed relays 37 through 48 (see Figure VI-14) are checked in the fill and print routines. Sticking relays, or relays which do not make contact can be located by observing the form of any fill or print errors, then logically tracing the error to a particular relay (or relay driver). TABLE VI-9 shows relation between Flexowriter positions and relays 37 through 48.

Flexowriter Position for Filling	Fill Relay	Flexowriter Position for Printing	Print Relay
6	43	6	37
1	48	1	42
2	47	2	41
3	46	3	40
4	45	4	39
5	44	5	38

TABLE VI-9.

Relation between Flexowriter Positions and the Fill and Print Relays.

AIR FILTERS

There are two air filters on the underside of the computer cabinet. To remove these for inspection,* disconnect the air filter retaining strips by removing the three thumbscrews below diode boards 2 and 4 and the fuse panel. Slide the air filters out, drawer fashion, avoiding contact with the glass wool.

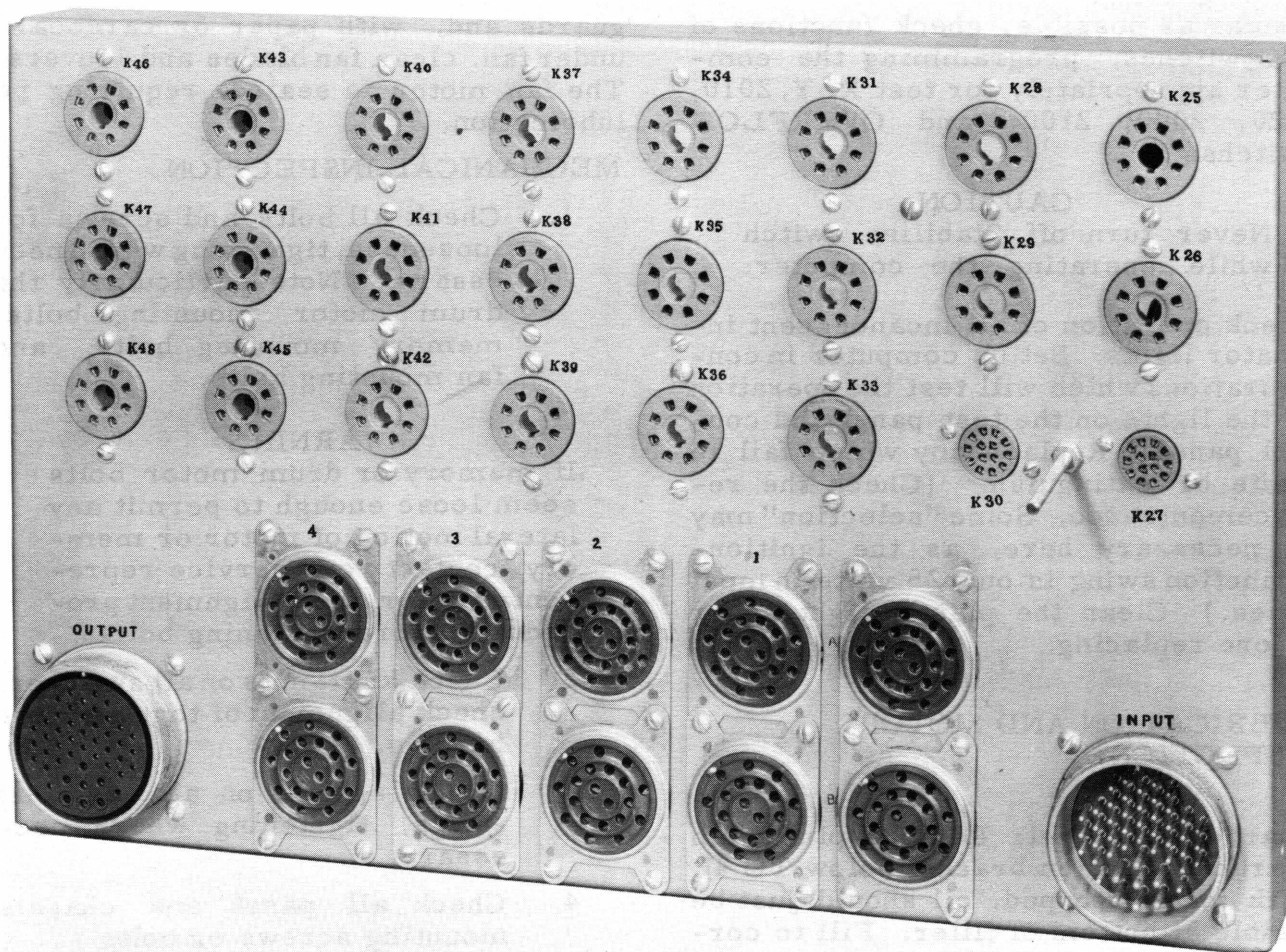


Figure VI-14. Upper Tube Chassis No. 2 (internal view)

NOTE

Do not attempt to clean filters. If dirty, replace with Warren and Bailey Dust Filters, size 16" x 25" x 1" or equivalent (NCR Component No. 274-0593).

CAUTION

Do not replace with any type of metal wool filter.

CONTROLS AND INDICATORS

With power remove from the computer, check all potentiometers manually for smoothness, making ohmmeter check on any having "rough spots". Reset to approximate position (R40 at full counter

clockwise position), then turn power on and adjust voltages to correct values according to instructions given under POWER SUPPLIES, "Voltages."

With power removed from the computer, check all toggle switches for "snap," replacing any which feel suspicious. Return all switches to normal position.

CAUTION

Be sure Stabiline switch is on before applying power.

Check all rotary switches for detent failures. Tighten any loose set screws, checking orientation of knobs, where applicable.

Insofar as possible, check functions of all switches, programming the computer appropriately for test X, Y, 2010, 2020, 2040, 2100, and OVERFLOW switches.

CAUTION

Never turn off Stabiline switch while operating the computer.

Check operation of all incandescent indicator lights. Set up computer in configurations which will test the operation of the lights on the test panel and control panel. Replace any which fail to ignite or extinguish. (Check the replacements, too. Some "selection" may be necessary here, as the ignition-extinction swing is only 25 volts in most cases.) Clean the plastic light cover before replacing.

LUBRICATION AND MOTOR INSPECTION

Gearbox Oil Level: Remove plug from gearbox filler (on bracket below TC 4). With motor stopped, oil should just be visible at bottom of filler. Fill to correct level with SAE 20W automotive oil (straight mineral oil - Pennsylvania base).

NOTE

Manufacturer recommends draining gearbox, with motor warmed up, every 1500 hours operating time. If desired, commutator and brushes may be checked and cleaned at this time. Clean commutator, if necessary, with crocus cloth only. If brushes are shorter than 1/2" length, replace with Stackpole 15L (available from Master Electric Co.).

Check operation of air louver motor (top of cabinet) and air louvers. Apply one drop of machine oil to each pivot point in louver assembly. Wipe louver chain with an oily finger. Remove fan

guards and, with paper or cardboard under fan, clean fan blades and louvers. The fan motor is sealed, requiring no lubrication.

MECHANICAL INSPECTION

1. Check all bolts and screws for looseness, tightening where necessary. Note particularly the drum motor mounting bolts, memory mounting bolts, and fan mounting bolts.

WARNING

If memory or drum motor bolts seem loose enough to permit any lateral motion of motor or memory, contact your service representative for shaft alignment procedure before tightening bolts.

2. Check knurl nuts on all switches, check alignment of toggle switch handles.
3. Check screws on all terminal strips, tightening where necessary.
4. Check all panel and chassis mounting screws or bolts.
5. Check caps over indicator lights; all should be finger tight.
6. Check all connectors, tightening any loose parts or ringnuts.
7. Check cables for wear or kinks.
8. Check all chassis for loose tubes, relays, or plug-in units.
9. See that all fuse holders are locked.
10. Generally examine frame, doors, and console for mechanical deficiencies of any kind.

COMPONENT TESTING

PLUG-INS

The following instructions for checking Alden plug-in units are based upon use of the Model 388 Computer Test Set

with the Model 389 special power supply, both manufactured by National Cash Register Company. In some instances, it should be noted that when a Model 388 Tester is used without the special power supply changes must be made in the test procedure.

Current practice for preventive maintenance checks on Alden plug-in units is to completely check all units every 15 weeks. This figure may vary for different installations as a result of variations in total running time, temperature conditions, etc. A vacuum-tube voltmeter with a minimum input impedance of 10 megohms should be used for voltage measurements. A 5 μ fd capacitor should be used in series with the oscilloscope probe. All units not conforming to the following test specifications must be rejected.

5687 and 5963 Drivers, Logical

Schematic Dwg. No.	Assembly Dwg. No.
5687: 002 63 002	002 67 001
5963: 004 63 002	004 67 001

Tube Test: Tubes are tested in the plug-in unit.

Plug unit into driver socket.

Set voltages to +225
+125
+100
-300

Set Test Switch to OFF.

Set Load Selector switch to OFF.

Set Clock Gate switch to 620K.

Set Clock Frequency switch to 110KC.

Set Clock switch to OFF.

Output: Driver socket pins 3 and 5.

STATIC TEST

- (1) Remove tube, set Test switch to Logical Driver, Static, and turn dc ON.
- (2) Observe voltage at the tube grid

pins (2 and 7). Voltage should be between +1.5v and +5.5v.

- (3) Observe voltage at output; this should be +125v.
- (4) Turn dc OFF and replace tube.

DYNAMIC TEST

- (1) Set Test switch to Logical Driver, Dynamic, and turn both dc and Clock ON.
- (2) Observe Driver output. Waveform should be a square wave of 25 volts amplitude (peak-to-peak) at clock frequency.
- (3) Check output for noise while tapping tube and plug-in. Reject noisy units.
- (4) Set Load Selector switch to Sum Load 56K for 5687 Driver, or to 47K for 5963 Driver.
- (5) Observe driver output. Waveform need not be square but the amplitude should be 25 volts.
- (6) For Production Test Limits: Set Load Selector switch to Product Load 31 ma for Driver 5687, or 6 ma for Driver 5963.

For Field Test Limits: Set Product Load switch to 25 ma for Driver 5687, or 4 ma for Driver 5963.
- (7) Depress Loading switch and observe driver output. Waveform should be a square wave of 25 volts amplitude. If amplitude is less than 25 volts, check tube emission.

5687 and 5963 Flip-Flop Drivers

Schematic Dwg. No.	Assembly Dwg. No.
5687: 003 63 002	003 67 001
5963: 005 63 002	005 67 001

Tube Test: Tubes are tested in the plug-in unit.

Plug unit into driver socket and plug a 001 67 001 flip-flop into flip-flop socket.

Set voltages to +225
 +125
 +100
 -300
 NCB = +6
 GCB = -6

Set Test switch to OFF.
 Set Load Selector switch to OFF.
 Set Clock Gate switch to 620K.
 Set Clock Frequency switch to 110KC.
 Set Clock switch to OFF.

Output: Driver socket pins 3 and 5.

STATIC TEST

- (1) Remove tube, set Test switch to Flip-Flop Driver, Static, and turn dc ON.
- (2) Observe voltage at the tube grid pins (2 and 7). Voltage should be between +34 and +40 volts.
- (3) Measure voltage at driver socket pins 3 and 5. This voltage should be +125 volts.
- (4) Turn dc OFF and replace tube in socket.

DYNAMIC TEST

- (1) Set Test switch to Flip-Flop Driver, Dynamic, and turn both dc and Clock ON.
- (2) Observe waveform at flip-flop plates (flip-flop socket pins 6 and 11) to insure flip-flop is cycling correctly at 1/2 clock frequency.
- (3) Observe driver output pins 3 and 5, waveform should be a square wave of 25 volts amplitude (peak-to-peak) at 1/2 clock frequency.
- (4) Check output for noise while tapping tube and plug-in. Reject noisy units.
- (5) Set Load Selector switch to Sum Load 56K for 5687 Driver, or to 47K for 5963 Driver.
- (6) Observe driver output. Waveform

need not be square but the amplitude should be 25 volts.

- (7) For Production Test Limit: Set Load Selector switch to Product Load 31 ma for Driver 5687, or 6 ma for Driver 5963.
 For Field Test Limits: Set Load Selector switch to Product Load 25 ma for Driver 5687, or 4 ma for Driver 5963.

12AT7 One-Shot, Unclamped
 Schematic Dwg.No. Assembly Dwg.No.
 105 63 001 105 67 002

Tube Test: Conduct standard tube emission test:

Production test emission limit:
 12 ma max; 6 ma min.

Field test emission limit:
 12 ma max; 6 ma min.

Plug unit into miscellaneous socket.
 Jumper clock to miscellaneous socket pin 8.

Set voltages to +225
 +125
 +100
 -300

Set Clock Frequency switch to 1KC.
 Set Clock switch to OFF.

Input: Miscellaneous socket pin 8.

Output: Miscellaneous socket pins 6 and 11.

STATIC TEST

- (1) Turn dc ON and measure the following voltages at miscellaneous socket.

pin 6 - +150
 pin 7 - + 25
 pin 10 - + 12
 pin 11 - +225

- (2) Observe output pins 6 and 11 while tapping tube and plug-in. Reject noisy units.

NOTE

Disregard Dynamic Test when using a Model 388 Tester without a special power supply.

DYNAMIC TEST

- (1) Turn Clock ON and observe output at pin 11. Trace will have positive pulses of approximately 100 volts amplitude (peak-to-peak) and 250 microseconds duration. Waveform will not be a good square wave. Frequency of pulses will be 1/2 clock frequency.
- (2) Observe output at pin 6. Trace will be the inverse of pin 11 output and the amplitude will be 60 to 100 volts (peak-to-peak).

5687 One-Shot, Clamped

Schematic Dwg.No.	Assembly Dwg.No.
106 63 028	106 67 029

Tube Test: Conduct standard tube emission test:

Production test emission limit:
56 ma max; 33.7 ma min.

Field test emission limit:
56 ma max; 30.0 ma min.

Plug unit into miscellaneous socket.

Set voltages to +225

+125

+100

-300

GCB = -8

Input: Miscellaneous socket pin 9.

Output: Miscellaneous socket pin 6.

STATIC TEST

- (1) Turn dc ON and measure the following voltages at miscellaneous socket:

pin 6 -	+100
pin 11 -	+225
pin 7 -	+ 27
- (2) Observe output at miscellaneous

socket pin 6 while tapping tube and plug-in. Reject noisy units.

- (3) Jumper GCB to miscellaneous socket pin 7 and observe miscellaneous socket pin 6 while tapping tube and plug-in. Reject noisy units.
- (4) Measure voltage at miscellaneous socket pin 6. This voltage should be +125.
- (5) Check back resistance of input diode D-1.

DYNAMIC TEST

(none)

12AT7 and 12BH7 Flip-Flops, Clamped	
Schematic Dwg.No.	Assembly Dwg.No.
12AT7: 202 63 011	202 67 012
12BH7: 202 63 001	202 67 002

Tube Test: Tubes are tested in the plug-in units.

Plug unit into flip-flop socket.

Jumper the following:

Flip-Flop socket pin 18 to driver socket pin 18.

Flip-Flop socket pin 6 to driver socket pin 3.

Flip-Flop socket pin 11 to driver socket pin 5.

Set voltages to +225

+125

+100

-300

NCB = +6

Set Test switch to OFF.

Set Load Selector switch to OFF.

Set Clock Gate switch to 100K.

Set Frequency switch to 110KC.

Set Clock switch to OFF.

Input: Flip-Flop clock gate - flip-flop socket pin 8 and 9.

Flip-Flop grids - flip-flop socket pins 7 and 10.

Output: Flip-Flop plates - flip-flop socket pins 6 and 11.

STATIC TEST

- (1) Measure plate voltages at flip-flop socket pins 6 and 11 with the flip-flop in each of its two stable states. These voltages should be +125 volts high output, and +100 volts low output.
- (2) Measure grid voltages with the flip-flop in each of its two stable states. E maximum should be approximately 0 volts and E minimum should be approximately -8 volts or more negatively. E minimum must not be more positive than -8 volts.
- (3) Measure voltages at junction of D1 and R7 while triggering the flip-flop manually. The higher of the two voltages should be greater than +5 volts. Repeat this test at the junction of D2 and R8.
- (4) Check flip-flop stability by leaving unit in each of its two stable states for at least 15 seconds. Tap tube and plug-in while flip-flop is in each state and reject units that indicate instability.

DYNAMIC TEST

- (1) Turn Clock ON and observe plate output. Waveform should be a square wave at 1/2 clock frequency.
- (2) Increase NCB to +10 volts. Flip-Flop should continue to operate.
- (3) Reset NCB to +6 volts.
- (4) Observe output while tapping tube and plug-in. Reject noisy units.
- (5) Set Load Selector switch to Sum Load 68K and observe flip-flop output. Waveform should have an amplitude of 25 volts but may not be a square wave.
- (6) Set Load Selector switch to Product Load 6 ma for 12AT7 Flip-Flop, or 8 ma for 12BH7 Flip-Flop. Waveform should have an amplitude of 25 volts and be a square wave. If

amplitude is less than 25 volts, Check tube emission. Set Product Load switch to 4 ma for 12AT7 Flip-Flop, or 6 ma for 12BH7 Flip-Flop when using field test limits.

12AT7 Flip-Flop, Unclamped
Schematic Dwg.No. 202 63 178 Assembly Dwg.No. 202 67 179

Tube Test: Tube is tested in plug-in.

Plug unit into flip-flop socket and plug a 003 67 001 Driver into driver socket.

Set voltages to +225
+125
+100
-300
NCB = +6
GCB = -6

Set Test switch to Flip-Flop Driver, Dynamic.

Set Load Selector switch to OFF.

Set Clock Gate switch to 110KC.

Set Clock switch to OFF.

Input: Flip-Flop socket pins 8 and 9.

Output: Flip-Flop socket pins 6 and 11.

STATIC TEST

- (1) Measure voltage at flip-flop socket pin 6. Trigger flip-flop to opposite state (if neon light glows touch flip-flop socket pin 10; if neon light is off, touch flip-flop socket pin 7) and note voltage again. Low reading should be between +85 and +120 volts. High reading should be between +191 and 205 volts.
- (2) Repeat above test measuring voltage at flip-flop socket pin 11.
- (3) With the flip-flop in each of its two stable states, measure the voltage at the junctions of D1, R7, and C3; and D2, R8, and C4. The higher voltage at both points should be greater than +5 volts.
- (4) Check flip-flop stability by leaving unit in each of its two stable states

for at least 15 seconds. Tap tube and plug-in while flip-flop is in each state and reject units that indicate instability.

DYNAMIC TEST

- (1) Turn Clock switch to ON and set Clock Gate switch to 620K.
- (2) Observe flip-flop output at flip-flop socket pins 6 and 11. Adjust GCB more negatively until flip-flop just ceases to operate. The GCB setting should be -9 volts or more negatively.
- (3) Reset Clock Gate switch to 270K and reset GCB to -6.5 volts. Disregard steps 4 and 5 when using Model 388 Tester without special power supply.
- (4) Set Clock Frequency switch to 10KC and increase the +100 volt supply slowly so that the difference between the two clamp voltages (+125 and +100 volts) decreases. The minimum voltage difference that will trigger each grid of the flip-flop should be 6 volts or more.
- (5) Reduce the voltage difference from some value above 10 volts to 3 volts. If the neon indicator is glowing trigger the flip-flop manually to turn the neon off. Increase the voltage difference gradually until the neon indicator begins glowing. The difference should now be 7 volts or more.
- (6) Reset the +100 volt supply to +100 volts and observe flop-flop socket pins 6 and 11 for noise while tapping tube and plug-in. Reject noisy units.

5963 Delay Line and Cathode Follower
Schematic Dwg.No. Assembly Dwg.No.
127 63 156 127 67 157

Tube Test: Conduct standard Gm and short tests.

Plug unit into miscellaneous socket.

Set voltages to +225
-300

Input: Miscellaneous socket pin 9.

Output: Miscellaneous socket pins 3 and 5.

STATIC TEST

(none)

DYNAMIC TEST

NOTE

When using a Model 388 Tester without the special power supply, the input to miscellaneous socket pin 9 should be a 110KC square wave clock instead of the sine wave signal. The output signals should be approximate square waves with amplitudes of 20 volts at pin 5 and 18 volts at pin 3.

- (1) Turn dc ON and connect a 110KC sine wave signal of 50 volts peak-to-peak amplitude to miscellaneous socket pin 9.
- (2) Observe outputs. Waveform at pin 5 should be a sine wave with a minimum amplitude of 40 volts peak-to-peak. Waveform at pin 3 should also be a sine wave with a minimum amplitude of 37 volts peak-to-peak.

5965 Special Dual Cathode Follower, Logical

Schematic Dwg.No. Assembly Dwg.No.
202 63 395 202 67 396

Tube Test: Conduct standard tube emission test.

Production test emission limit:
12.0 ma max; 6.0 ma min.

Field test emission limit:
12.0 ma max; 5.6 ma min.

Plug unit into cathode follower socket.

Set voltages to +225
-300

Set CF Cathode Resistor switch to 39K.
Set Clock Frequency switch to 110KC.
Set Clock switch to OFF.

Input: Pins 7, 8, 9, and 10 (wired internally to clock).

Output: Pins 3, 5, 6, and 11.

STATIC TEST

- (1) Turn dc ON and measure voltage at each of the output pins. These voltages should be +125 (+1 -0) volts.
- (2) Depress Static Clock switch and repeat measurement of output pins. These voltages should be +100 (+2 -0) volts.

DYNAMIC TEST

- (1) Turn Clock switch to ON and observe output pins. Output waveform should be identical to clock waveform except that amplitude will be approximately 3% less.
- (2) Check outputs for noise while tapping tubes and plug-in. Reject noisy units.

12AT7 Dual Cathode Follower
Schematic Dwg.No. Assembly Dwg.No.
202 63 009 202 67 010

Tube Test: Conduct standard tube emission test.

Production test emission limit:
12.0 ma max; 6.0 ma min.

Field test emission limit:
12.0 ma max; 5.6 ma min.

Plug unit into cathode follower socket.

Set voltages to +225
+125
+100
-300

Set CF Cathode Resistor switch to 39K.
Set Clock Frequency switch to 110KC.
Set Clock switch to OFF.

Input: Cathode Follower socket pins 7, 8, 9, and 10.

Output: Cathode Follower socket pins 3, 5, 6, and 11.

STATIC TEST

- (1) Turn dc ON and measure output voltage at output pins. These voltages should be +125 (+1 -0) volts.
- (2) Depress Static Clock switch and repeat measurement of output pins. These voltages should be +100 (+2 -0) volts.
- (3) Observe outputs for noise while tapping tube and plug-in. Reject noisy units.

DYNAMIC TEST

- (1) Turn clock ON and observe outputs. Waveform should be identical to clock at pin jack C except that amplitude will be about 5% less.

5881 Cathode Follower Driver
Schematic Dwg.No. Assembly Dwg.No.
202 63 292 202 67 293

Tube Test: Conduct standard tube emission test.

Production test emission limit:
41 ma min.

Field test emission limit:
33 ma min.

No static or dynamic tests of this plug-in unit are possible on the plug-in tester.

Measure resistance between Alden plug and tube socket pins.

6BQ7-A Dual Relay Driver
Schematic Dwg.No. Assembly Dwg.No.
126 63 101 126 67 102

Tube Test: Conduct standard tube emission test.

Production test emission limit:
21.6 ma max; 12.0 ma min.

Field test emission limit:
21.6 ma max; 11.0 ma min.

Plug unit into miscellaneous socket.

Set voltages to +125
+100
-300

Input: Miscellaneous socket pins 7, 8, 9, and 10.

Output: Miscellaneous socket pins 3, 5, 6, and 11.

STATIC TEST

- (1) Remove tubes from plug-in and jumper +125 volts to miscellaneous socket pins 7, 8, 9, and 10.
- (2) Turn dc ON and measure voltage at tube socket pins 2 and 7 of both tubes. These voltages should be between +1.5 and 5.5 volts.
- (3) Move jumper to input pins from +125 to +100 volts and repeat measurement of above voltages. These voltages should now be between -11 and -19 volts.

DYNAMIC TEST

(none)

6BQ7-A Clipper

Schematic Dwg.No.	Assembly Dwg.No.
106 63 007	106 67 017

Tube Test: Conduct standard tube emission test.

Production test emission limits:
21.6 ma max; 12.0 ma min.
Field test emission limit:
21.6 ma max; 11.0 ma min.

Plug unit into miscellaneous socket.

Jumper the following pins on the miscellaneous socket:

pin 13 to +20 volts.
pin 15 to NCB.
pin 5 to driver socket pin 5.
pin 3 to driver socket pin 3.

Set voltages to -300
+225
+125
+100
+20
NCB = +10

Set Test switch to OFF.
Set Load Selector switch to OFF.

Input: Miscellaneous socket pins 8 and 9.

Output: Miscellaneous socket pins 3 and 5.

STATIC TEST

Turn dc ON and measure voltage at miscellaneous socket pins 3 and 5. Voltages should be +100 volts.

NOTE

Disregard DYNAMIC TEST when using Model 388 Tester without special power supply.

DYNAMIC TEST

- (1) Connect a 50 volt (peak-to-peak) 55KC sine wave to miscellaneous socket pin 8 and observe square wave output at miscellaneous socket pin 5. Output waveform should have an amplitude of 25 volts.
- (2) Reduce sine wave input signal gradually until output signal ceases to appear. Input sine wave should, at this time, have an amplitude of less than 20 volts peak-to-peak.
- (3) Disconnect input sine wave signal from miscellaneous socket pin 8 and connect to pin 9.
- (4) Observe output waveform which should have an amplitude of 25 volts, at miscellaneous socket pin 3 and repeat step (2) above.

5687 Dual Phase Inverter

Schematic Dwg.No.	Assembly Dwg.No.
106 63 006	106 67 016

Tube Test: Conduct standard Gm and short test.

Plug unit into miscellaneous socket.

Set voltages to +225
-300

Input: Miscellaneous socket pins 8 and 9.

Output: Miscellaneous socket pins 3, 5, 6, and 11.

STATIC TEST

Turn dc ON and, with no signal connected to the input, observe all four outputs for noise while tapping tube and plug-in. Reject noisy units.

NOTE

Disregard Dynamic Test when using Model 388 Tester without the special power supply.

DYNAMIC TEST

- (1) Connect a 60 volt (peak-to-peak) 55KC sine wave to input pin 3 and observe output signals at miscellaneous socket pins 5 and 6. Amplitude of the output signals should be approximately 50 volts peak-to-peak, and should be equal within 2% of each other. Waveform at pin 6 will be in phase with the input signal and waveform at pin 5 will be 180 degrees out of phase with the input signal.
- (2) Connect above input signal to miscellaneous socket pin 9 and observe output signals at miscellaneous socket pins 3 and 11. Amplitude of the two waveforms should be approximately 30 volts and equal within 2% of each other. Waveform at pin 11 will be in phase with the input signal and waveform at pin 3 will be 180 degrees out of phase with the input signal.

12AT7 Schmitt Trigger

Schematic Dwg.No.	Assembly Dwg.No.
106 63 008	106 67 018

Tube Test: Conduct standard Gm and short test.

Plug unit into miscellaneous socket.

Jumper miscellaneous socket pin 18 to +225.

Set voltages to -300
+225

Set Sine Wave Oscillator switch to 110KC.

Input: Miscellaneous socket pin 6.

Output: Miscellaneous socket pins 3, 5, and 6.

STATIC TEST

Turn dc ON and measure the following voltages at miscellaneous socket.

pin 3 - +65
pin 5 - +225
pin 6 - +225

NOTE

Disregard Dynamic Test when using Model 388 Tester without special power supply.

DYNAMIC TEST

Jumper 40 volt (peak-to-peak) 110KC sine wave to miscellaneous socket pin 8 and observe output at miscellaneous socket pin 6. Waveform should be a square wave with an amplitude of 60 volts (peak-to-peak), and should be at clock frequency.

Quintuple Input Coupler

Schematic Dwg.No.	Assembly Dwg.No.
202 63 013	202 67 014

Plug unit into miscellaneous socket.

Jumper clock to miscellaneous socket pins 7, 8, 9, 10, and 13. Jumper miscellaneous socket pins 3, 5, 6, 11, and 18 to ground.

Input: Miscellaneous socket pins 7, 8, 9, 10, and 13.

Output: Grounded.

STATIC TEST

Turn dc ON and measure voltages at junction of resistor, diode, and capacitor for each of the line sections of the plug-in. These voltages should be between +5.3 and +6.0 volts. If any voltage is less than +5.3 volts, check back resistance of diode.

NOTE

Disregard Dynamic Test when using Model 388 Tester without special power supply.

DYNAMIC TEST

- (1) Turn clock ON and observe waveform at junction points given in STATIC TEST. Waveform should be a series of alternate positive and negative pulses.
- (2) Reduce the NCB to zero volts while continuing to observe the waveforms. The negative pulses should be clipped and the positive pulses should remain unchanged.

TUBES

Tests for tubes in plug-in units are given on the appropriate plug-in unit test procedure. In many cases, the tube is tested in the plug-in unit. Tests for tubes not in plug-in units can be made on any good quality commercial tube tester. Such tubes include the following types: OA2, OB2, 2D21, 5U4G, 6AU6, 6BN6, 6L6G, 6SJ7, VR-150, 6080.

DIODES

The Model 102-A and Model 126 utilized NCR diode Types A through G. TABLES VI-10, VI-11, and VI-12 give the discard and acceptance specifications for these types. The discard specifications should be used to determine end of useful life during preventive maintenance testing. Also since Types A, B, C, and D are now obsolete, they should be replaced, when they fail, by the following types:

1. Type A is replaced by Type F.
2. Types B or D are replaced by Type G_1 in the ratio of one Type G_1 to two Type B or D.

NOTE

Type G_1 must not be operated in parallel with Types B or D. When replacing the latter with Type G_1 , all parallel diodes should be replaced at the same time.

Type G_1 can also be used as a replacement for Type F_1 which is used in some plug-ins.

3. Type C is replaced by Type E.

The temperature derating specification, one microampere per degree centigrade above 25° C (77°F), should be applied as a correction when testing diodes at temperatures above 25° C.

Any A, B, C, D, and E diode can be regraded as any other Type A through E, provided the diode is re-tested to the specifications of the other type. However, Types A, B, C, D, and E cannot be re-tested to F or G specifications, as F and G are "gold-bonded" types of different manufacture.

Type	Back Current Max. at 25°C	Forward Voltage Max. at 25° C
A	70 microamps at -25v	4.0v at 30ma
B	150 microamps at -25v	3.0v at 30ma
C	Replaced by Type E	
D	150 microamps at -25v	2.2 to 2.7v at 30 ma
E	7 microamps at -5v 14 microamps at -10v	1.5v at 4ma
F, F_1	60 microamps at -25v	1.0v at 15ma
G, G_1	300 microamps at -25v	1.0v at 50ma
NOTE: Temperature derating: 1µa per degree centigrade		

TABLE VI-10
Diode Discard Specifications

Type	Application	Back Current at 25°C	Forward Current at 25°C	Minimum Peak Inv. Voltage
A GE No. G-5S17	Logical Networks Low Back Current Circuits	0-40 μ a at 25v	30ma at 0-3v	70v
B GE No. G-5S18	Low Forward Voltage Circuits Medium Current Clamping Circuits	0-100 μ a at 25v	30ma at 0-2.5v	70v
C D*	Replaced by Type E Selected from Type B for High Current Clamping Circuits (requiring diodes in parallel)	0-100 μ a at 25v	30ma at 2.2 - 2.5v	70v
E GE No. G-5S41	Low Voltage Clamping Circuits Coupling Circuits Low Voltage Rectifiers	0-5 μ a at 5v 0-10 μ a at 10v	Greater than 4ma at 1v	50v

* Type D selected from G-5S17, G-5S18, or G-5S41
Stability Requirements for A, B, and D diodes:

1. These diodes shall be checked by applying -25v, 60 cycle, sinusoidal voltage. Ambient temperature not to exceed 35°C.
2. The back current versus reverse voltage characteristic, as observed on an oscilloscope, shall show no more than 1.0 μ a of hysteresis or flutter.
3. No increase in back current with time shall be allowed.
4. Temperature derating shall be one microampere per degree of Centigrade.

NOTE: The above Types are all manufactured by General Electric to NCR specifications.

TABLE VI-11
 Acceptance Specifications for Old Type Diodes.

Type	Application	Maximum Reverse Current		Minimum Forward Current	Minimum P. I. V.
		25°C	45°C*	25°C	25°C
F (S-125)	Logical Networks	40µa at -25v	80µa at -25v	20ma at +1v	70
F ₁	Med. Current Clamping Circuits with Leads	40µa at -25v	80µa at -25v	20ma at +1v	70
G (S-158)	High Current Clamping Circuits	200µa at -25v	400µa at -25v	70ma at +1v	70
G ₁ (S-158)	High Current Clamping Circuits with Leads	200µa at -25v	400µa at -25v	70ma at +1v	70

*97.5% of diodes meeting reverse current requirements at 25°C will meet this requirement.

General Requirements:

1. Maximum 1.0µa hysteresis at 25v, 60cps.
2. No apparent flutter or creep.
3. Maximum reverse current specifications shall be corrected by an increase of one microampere per degree centigrade for ambient test temperatures from 25°C to 40°C.

NOTE: Above types are all manufactured by Transitron to NCR specifications. The Transitron number is listed above for each type.

TABLE VI-12
Acceptance Specifications for New Type Diodes.

AUXILIARY EQUIPMENT

MODEL 126 MAGNETIC TAPE UNIT

The Tape Unit Read-Write Program (coding given in APPENDIX E) has two major sections. In the first section, a complete set of addresses can be written, read, and checked. The last address in the set is specified by m₂ of cell 0115 (777 in the coding given).

The "write addresses" portion automatically writes a maximum address (17777) after the last address called for in cell 0115. The "read addresses" portion searches to block 00000, then reads and checks addresses in sequence up to and including the address called out in m₂ of 0115.

The second section writes a test number in every block on the tape in Mode 1, reads back and checks this informa-

tion, then writes the test number in every block in Mode 2, and reads back and checks.

To use the program, first clear the memory with the "Erase Main Memory" program (see APPENDIX E), then fill the Tape Unit Read-Write program. Filling may be checked by taking a memory sum and shifted memory sum with the programs given under "Memory Sum" and "Shifted Memory Sum" (see APPENDIX E) before starting the program.

The sequence of operations is as follows:

- 1) Types "Write Addresses"
Writes addresses 0,0,1,2,....
777, 1777
Types "00 0001 7777 0000".
 - 2) Types "Read Addresses"
Searches to address zero and reads and compares the addresses just written. If error, prints correct address, then incorrect address.
Types "00 0000 1000 0000".
 - 3) Types "Write Mode 1"
Searches to address zero and writes a one-zero pattern, Mode 1, in all blocks.
Types "00 0000 1000 0000".
 - 4) Types "Read Mode 1"
Searches to address zero and reads and compares each block, Mode 1. When there is an error, the block address, correct cell contents, and actual contents of the eight cells of Mode 1 are printed. These eight cells are numbered 0 to 7 in the second digit of the word in the cell. Example: 15 0360 7417 0360, cell number 5 should contain the digit 5 in the position shown.
Types "00 0000 1000 0000".
 - 5) Types "Write Mode 2"
Searches to address zero and writes a one-zero pattern, Mode 2, in all blocks.
Types "00 0000 1000 0000".
 - 6) Types "Read Mode 2"
Searches to address zero and reads and compares each block, Mode 2, as described in Step 4.
Types "00 0000 1000 0000".
 - 7) Repeats Steps 3, 4, 5, and 6, using a zero-one pattern (cell contents equal 1x 7417 0360 7417).
 - 8) Returns to read addresses in Step 2.
- Test switches are used as follows:
- 2010 - DOWN: Halt at end of read or write section.
UP: Continue to next section.
- 2020 - DOWN: Iterate
UP: Halt immediately. Restore by pressing Compute button.
- 2040 - DOWN: Do not repeat command.
UP: Repeat command with check and print if there is an error (applies to "read" sections only).
- 2100 - DOWN: Do not repeat command.
UP: Repeat command without check.
- To run the complete program, throw switches as follows: 2010 UP, 2020 DOWN, 2040 DOWN, 2100 DOWN.
- The entire routine starts in cell 0100, but sections of the routine can be used separately. To start computation, type 0100f hyphen - then press Compute button.

IBM EQUIPMENT

General

Ordinarily, lease of IBM equipment includes regular preventive maintenance to that equipment. In case of trouble definitely traced to the IBM equipment, call the nearest IBM representative.

As mentioned previously, IBM machines when used as input or output devices with a Model 102-A General Purpose Computer must be modified.

The NCR number is provided only for reference. In order to secure the desired modifications, it is necessary to notify IBM Corporation that modification is required. The modification will be made by IBM either before or after delivery of the machine. The modification will have no effect upon the normal service provided by IBM.

Operational Testing

The IBM Test Program (see APPENDIX E for coding) will punch a deck of 25 IBM cards (10 octal, 15 decimal) using every row and column at least 5 times. In the second part of the program, these cards are read and checked. The purpose of the program is to check the operation of the IBM circuits in the computer in conjunction with the IBM punch and reader.

The procedure for filling the program is as follows: (see APPENDIX E for coding of all programs)

1. Clear the memory with the Erase Main Memory routine.
2. Fill IBM Test program, Part I.
3. Fill Memory Sum and Shifted Memory Sum routines.
4. Turn on the IBM punch.
5. Insert a deck of at least 27 blank cards in the hopper.
6. Press the IBM start button once.
7. Type "1700f-". Then press Compute button.

NOTE

The results will check only if the reader plugboard is wired to transmit brush propositions to the computer in the same order as for punching.

8. Press the start button on the IBM punch to obtain the last card of the deck.
9. Transfer the deck to the IBM reader hopper.
10. Press the IBM start button once to feed the first card.
11. Clear the memory as in Step 1.
12. Fill Part II of IBM Tests.
13. Repeat Step 3.
14. Repeat Step 7.

Printouts:

Error:

00 0000 0000 00 (octal number of card with error). Next four words are contents of error card, as read into the buffer register.

Correct:

Octal
No.

Contents

00-04	00000000000000	11111111111111	22222222222222	33333333333333
05-11	04444444444444	15555555555555	26666666666666	37777777777777
12-16	+000000000	p111111111	-22222222	n333333333
17-23	+444444444	p55555555	-66666666	n77777777
24-30	+888888888	p99999999	-88888888	n99999999

Glossary

These definitions include many of the words and phrases most commonly used in "computer language", with specific application to the 102-A Computer.

Access Time - The time required for a certain cell on the surface of the rotating memory drum to begin to be carried under a read, write, or read-write head after the computer has called for that cell.

Adder - A device which can form the sum of two or more numbers, or quantities, impressed upon it.

Address - The number specifying the position of a particular cell on the memory drum; or the position of a particular block on the tape of a magnetic tape unit.

Address Channel - See "Word Channel".

Address Section - That section of a command consisting of the least significant 36 bits of the word, octal digit positions 0_0 through 0_{11} , and containing the addresses of the command (usually three addresses per command).

Alarm - An indication of improper programming or certain types of improper machine operation.

"And" - See "Logical Product".

Arithmetic Element - That part of the computer which performs arithmetic calculations.

Arithmetic Instruction - One of the instructions which result in the performance of specific arithmetic operations by the computer. See Command Code, page III-6.

Arithmetic Time - The time required to execute an arithmetic operation after the operands have been located on the drum.

Arithmetic Unit - See "Arithmetic Element".

Base (or Radix) - The integer of whose successive powers the digits of a number are the coefficients.

Binary Cell - An information-storing element which has two stable states.

Binary-Coded Decimal System - A system of number representation in which the decimal digits of a number are expressed by binary numbers.

Binary Digit - A digit of a binary number; either 0 or 1, or the content of a binary cell.

Binary Number System - The number system with the base two, and using only the two digits 0 (zero) and 1 (one).

Binary Point - The radix point in the binary system.

Bit - A binary digit.

Bi-Stable Element - A multivibrator, flip-flop, or toggle circuit having two

stable states and ordinarily two input terminals, or two types of input signals, each of which corresponds to one of the two states. This circuit remains in either state until caused to change to the other state by application of the corresponding signal. See page IV-7.

Block - (1) A single process of the computer corresponding to one state of the program counter. (See Flow Diagram, Figure I in Volume II) (2) In the Model 126 Tape Unit, the arrangement of data on the magnetic tape in a group of 17 words.

Block Address - The address of a particular block on a magnetic tape. Blocks may be addressed from 00000 through 17,777 octally.

Block Search - The automatic searching process through which the Model 126 Tape Unit will proceed as a result of the BLOCK SEARCH command from the computer. The tape unit will search for a specific block independently of the computer, that is, while the 102-A is carrying out other commands.

Boolean Algebra - An algebraic system for manipulating symbolic logic. See page C-1.

Buffer Register - An eight-word delay line register, designated by the letter "J" in the 102-A. It acts as a buffer between the magnetic tape or the IBM cards and the main memory of the computer; and may also be used as an eight-word extension of the main memory. The addresses of the buffer cells are 2000 through 2007.

Bus - A wire or group of wires over which information coming from any of several sources is conveyed to any of several destinations.

Carry - (1) A condition occurring in addition when the sum of digits in the same column equals or exceeds the base of the number system in use. (2) The digit to be carried over to the next column. (3) The action of carrying over this digit.

Carry Flip-Flop - The flip-flop used to store the carry digit in any arithmetic operation. In the 102-A, the carry flip-flop is usually, but not always, designated by K.

Cathode Follower - An electronic tube and associated circuit, which may be considered as an in-phase amplifier with gain of less than unity and with output from the cathode, used for decoupling a load from an input to prevent loading of the latter.

Cell - A particular position in the main memory, in a register, or a portion of the magnetic tape which can contain one word; and which has a distinguishing address.

Channel - One of the horizontal rings around the circumference of the memory drum, so identified and placed as to be accessible to the appropriate electromagnetic head for reading, writing, or both.

Channel Selector Flip-Flops - The five flip-flops, L_1 through L_5 , that determine which channel of the memory is to be selected for look-up or putaway.

Check Problem - A problem whose incorrect solution indicates an error in the operation of the computer.

Clamp Diode - A diode used to limit the voltage swing at a given point in a circuit.

Clear - A control function generated by pushing the "Clear" button on the control panel. This control synchronizes the pulse and digit counters, sets the K flip-flop true, clears (turns off) the A_1 through A_6 logical flip-flops and R_1 and R_2 record flip-flops, and sets the program counter to rest (Block 0). A clear pulse occurs automatically as one of the last steps in the power supply sequence.

Clear Proposition - The logical proposition which is true when the "Clear" button on the control panel is pressed.

Clipper - A special amplifier, used after the preamplifier and phase inverter, to shape pulses coming from the memory drum.

Clock - The basic timing pulses of the computer, consisting of 42 timing pulses per word repeated 64 times per drum revolution and magnetically recorded in duplicate on two channels of the revolving memory drum. The clock rate in the 102-A Computer is approximately 107 kc.

Coding - The process of translating a program into the machine's language.

Column - Referring to an IBM card, a vertical arrangement of data representing one digit.

Command - Information which, when properly coded and introduced as a word into the computer, causes the latter to perform one of its operations. A command usually includes one or more (up to three) addresses, as well as the operation.

Command Code - The numbers representing the commands which the 102-A Computer is capable of performing. See page III-6.

Comparator - An amplifier, the output of which is proportional to the difference between two input voltages; used in the 102-A power supply.

Complement - A number whose representation is derived from the finite positional notation of another by one of the following rules:

- (a) True complement - Subtract each digit from the base less 1, then add 1 to the least significant digit, executing any carries required.
- (b) (Base - 1)'s complement - Subtract each digit from the base less 1.

Computer - A device which can accept information and supply information and in which the supplied output information is derived from the accepted input information by means of a process of logic

Conditional Transfer - A command which, depending upon some property of a given number or numbers, will cause the proper one of two other commands to be executed next.

Control Circuits - The circuits of the computer which effect the carrying out of commands in proper sequence, the interpretation of each command, and the application of the proper commands to the arithmetic element and other circuits in accordance with this interpretation.

Control Console - A metal desk on which are mounted the computer operator's control panel and a modified Flexowriter.

Control Number - The address of the next command to be used by the computer.

Control Register - That register on the

memory drum, but not part of the main memory, which contains the address of the next command to be used (Control No.), and which controls the dispatching of the word resulting from an operation to its proper destination after each command is executed.

Count - (As used in 102-A) - Movement of the program counter from one number to the next number in numerical order when the K (decision) flip-flop is set false.

Counter - A group of flip-flops logically connected to indicate a numerical sequence.

Counter Total Exit - A row of plug hubs on the IBM control panel. See page IV-52.

Decimal Number System - The method of positional notation using ten as the base.

Decimal Point - The radix point in the decimal system.

Decision Flip-Flop - The K flip-flop, which determines at the end of a block whether the program counter will count or skip or stick.

Delay Line - A device which provides a given amount of delay time (an integral No. of clock pulses) between input and output.

Delay-Line Register - An electronic delay line, one or an integral number of words long, together with input, output, and circulation circuits.

Digit - One of a definite set of characters used as coefficients of the base in a number system; usually refers to octal digit when used with the 102-A computer.

Digit Counter - A counter consisting of

logical networks, flip-flops and drivers which is driven by pulses from the pulse counter and is used to count octal digit positions in each word time. There are 14 digit positions, numbered 0_0 through 0_{13} , in each word of the 102-A.

Digital Computer - One in which information, numerical or otherwise, is represented by means of combinations of characters in such a way that the number of distinguishable combinations is much greater than the number of distinguishable characters. Thus a digital computer is one which makes explicit use of a language or set of symbols.

Diode - An electronic component having a high impedance to current flow in one direction, and a low impedance to current flow in the opposite direction.

Diode Board - One of the four large boards (numbered 1 through 4 and placed two on each of the long sides of the 102-A Computer) used for holding the diodes of the logical networks. Total capacity of the four boards is 5200 diodes. See page I-14.

Double-Length Putaway - The putaway of a two-word result in the memory of the 102-A. The operations "md" (multiply double-length); "dr" (divide with remainder); and "sf" (scale factor) require a double-length putaway.

Double-Precision Number - A number having twice as many significant digits as are ordinarily used in the computer. In the 102-A, this means 72 binary digits, since 36 bits from each of two words are used for the magnitude of a double-precision number.

Driver - A D-C coupled amplifier connected to the output of a flip-flop or a logical network to supply sufficient cur-

rent to drive large logical networks. The driver also performs the logical operation of negation.

✓ Drum - The information storage device or memory of the computer, consisting of a cylinder with a magnetic surface, revolving on its vertical axis, and on which information is recorded or from which information can be played back in the form of positive or negative pulses by means of electromagnetic write or read heads. See page I-12,

Dummy - A program counter block where nothing is changed except, in some cases, the state of the K (decision) flip-flop to prepare for the next operation.

Eleven (11) Row - See "X Row" as applied to IBM cards.

Erase Head - One of five electromagnetic elements used to erase information contained in the E, F, G, H, or J registers of the memory drum.

E Register - See "Register" and page IV-19.

Error - A case of improper programming or possible improper computer operation which causes incorrect results.

Extract - A logical command to the computer which moves selected bits from one word into the same positions of another word. See page III-10.

False - (1) State of a proposition when the voltage is at the lower of two logical reference levels; e.g. 100 volts in the 102-A Computer. (2) State of a flip-flop which will make its associated output proposition false; also called its "zero" or "off" state.

Fixed-Point System - A system of number notation in which a number is represented by a single set of digits in which the radix point is not numerically expressed, but is understood.

✓ Flexowriter - The trade name of an automatic electric typewriter manufactured by Commercial Controls Corporation and used as an input-output device with the 102-A Computer. It is slightly modified and mounted on the control console.

Flexowriter Relays - The relays used for opening and closing the circuits between the Flexowriter and the computer.

Flexowriter Sum - The logical sum of several signals from the Flexowriter decoding matrix. See page IV-42.

Flip-Flop - A bi-stable multivibrator. See "Bi-Stable Element" and page IV-7.

Flow Diagram - A diagram showing the sequence of all the computer internal processes as controlled by the program counter. Each process is shown in a block.

F Register - See "Register" and page IV-19.

Function - The activity resulting from a command.

Gate - (1) A circuit having an output and a multiplicity of inputs so designed that the output is energized when and only when a certain definite set of input conditions are met. (2) A signal used to permit the passage of other signals through a circuit.

Go Ahead - (1) The switch circuit in the Flexowriter which supplies a signal to the computer indicating that the Flexowriter has completed its operation and

therefore the computer can initiate the printing of the next character. (2) The proposition supplied to the computer by this circuit.

G Register - See "Register" and page IV-19.

Grid Clamp Bias - A voltage applied through a diode used to clamp the grid of a flip-flop at about -9v.

Grid Proposition - The input to a flip-flop, usually a product of logical propositions and clock pulses.

Halt - A command which returns the computer to rest.

H Register - See "Register" and page IV-19.

Hyphen (Key on Flexowriter) - A control symbol on the Flexowriter which results in shifting information in the computer from the E (input) register to the H (control) register. The hyphen symbol is on the bottom row of the Flexowriter keyboard and should not be confused with the minus sign appearing on the top row.

Idle - The condition of the 102-A Computer in Block 0 (zero) when the computer is turned on, with all voltages applied, and is waiting for an input signal from either an appropriate Flexowriter key or the "Compute" button on the control panel. Same as "Rest".

Input - Information received by the computer that is either to be processed by the computer or used to control operations within the computer.

Input Devices - Units used to convert alphabetical or numerical information in written or coded form into electronic pulses for entry into the computer. Present input devices include the modi-

fied Flexowriter, Model CRC 126 Magnetic Tape Unit, and modified IBM Type 523 Summary Punch.

Input-Output Command - One of a series of commands which results in reading information from one of the input devices into the computer, or taking information from the computer and delivering it to one of the output devices. See command Code, page III-6.

Input Register - The E recirculating register which accepts individual digits or symbols from the Flexowriter.

Instruction - Same as "Command".

J Register - Same as "Buffer Register".

Latent Time - The time used by the computer in internal operation other than arithmetic or access times.

Line - See "Delay Line".

Logical Product - A function in symbolic logic; all terms of the input must be true for the product to be true. See page D-1.

Logical Sum - A function in symbolic logic; at least one term of the input must be true for the sum to be true. See page D-1.

Logical Command - One of a series of commands which results in the performing of logical operations by the computer. See Command Code, page III-6.

Look-up - The computer operation of finding a command or number in a specified memory cell. See also "Block Search", page IV-58.

Lower Clamp - A diode used to limit the voltage swing of a certain circuit point at the lower of two reference lev-

els, namely +100 volts in the 102-A.

LSBD - Least significant binary digit. The bit which is the coefficient of the lowest power of 2 present in the binary number.

LSD - The least significant digit. The digit which is the coefficient of the lowest power of ten present in the decimal number.

LSOD - Least significant octal digit. The digit which is the coefficient of the lowest power of 8 present in the octal number.

Magnitude - The size of a number, without reference to its algebraic sign.

Magnitude Section - That section of a number-word consisting of the least significant 36 bits of the total 42 bits, and containing the magnitude of the number.

Main Memory - The 16 channels in the main memory section of the drum. These channels contain 1024 cells identified by octally-numbered addresses from 0000 to 1777. The first two octal digits, 00 to 17, specify the channel number; while the last two digits, 00 to 77, specify the sector number.

Matrix - A network of diodes and resistors combined in such a way so as to generate a set of outputs, each of which corresponds to a specific combination of inputs. For example, the matrices used to generate the pulse and octal digit propositions: see Figure 7, Volume II.

Memory - The storage facility within the computer, in the form of a magnetic drum, into which information can be introduced and then extracted at a later time.

Memory Capacity - The maximum number of words which the memory can store. In the 102-A this total is 1024 words in the main memory plus 8 words in the buffer register.

Memory Flip-Flop - A flip-flop which follows a clipper and serves to convert the memory output to logical form with the proper timing.

Memory Tube Chassis - A chassis shaped like a cross and mounted directly above the memory drum in the center of the 102-A computer.

This chassis contains plug-in units and associated wiring for the record tubes and preamplifiers, which are closely associated electronically with the read, write, and erase heads on the drum.

Minimum Access Coding - The coding process which locates operands in such a way as to minimize their access time for either look-up or putaway. The result is faster operation of the computer in any given program of commands.

Mode - The manner or particular form in which an operation is performed; for example, the octal mode, decimal mode, or alphabetic mode.

MSBD - Most significant binary digit. The bit that is the coefficient of the highest power of 2 present in the binary number.

MSD - Most significant digit. The digit that is the coefficient of the highest power of ten present in the decimal number.

MSOD - Most significant octal digit. The digit that is the coefficient of the highest power of eight present in the octal number.

Negation - The change of a logical proposition from its existing condition to its opposite condition; i. e., true becomes false or false becomes true. See page D-3

Noise Clipper Bias - A voltage fed through a diode at the input to a flip-flop so as to make that flip-flop insensitive to noise pulses.

Non-Return to Zero - A term used to describe the fact that on the 102-A memory drum all areas of all cells are magnetized to saturation in one direction or the other once the drum is filled; and the logical propositions are either high or low during each clock period. Information is represented by continuous levels rather than by pulses.

"Not" - See "Negation" and page D-3.

Number - A word which is not a command. It is composed only of digits to signify sign and magnitude.

Number System - A system of representing magnitudes with ordered coefficients (digits) associated with powers of a given number called the base. Binary, octal, and decimal numbers (base 2, 8, and 10 respectively) are used with the 102-A Computer.

Octal - The number system with the base eight. The digits in this system are 0 through 7.

One-Shot - A special multivibrator circuit with one stable state which can be triggered into the unstable state by a pulse of the correct amplitude and polarity. It will return to the stable state after an interval depending upon the circuit constants.

Operand - A word on which an operation is to be performed.

Operation - A specific arithmetic, logical, transfer, input, or output function which the computer is capable of performing. The Operation (capitalized to distinguish from the common meaning) is generally coded in numerical form and appears as the two most significant octal digits of a command.

Operation Time - The total execution time of a command, which is the sum of the latent time, the access time, and the arithmetic time.

"Or" - See "Logical Sum" and page D-1.

Order - Same as "Command".

Output - Information produced by the computer under automatic control of the program.

X Output Devices - Units used to convert electronic pulses from the computer into numerical or alphabetical information in written or coded form. Present output devices include a modified Flexowriter, Model CRC 126 Magnetic Tape Handling Unit, and modified IBM Type 523 Summary Punch.

Overflow - (1) The condition that arises when the result of an arithmetic operation has exceeded the capacity of the arithmetic register, usually signaled by a one appearing as the LSB in the Operation portion (the sixth MSBD) of a word. (2) The carry digit arising from the overflow condition.

Parallel Transmission - See definition under "Serial Transmission".

P-Cams - The cams in an IBM unit used for timing the punching of holes in an IBM card.

Phase Inverter - A circuit following the preamplifier and used to invert signals from a read head prior to clipping.

Playback - The operation of obtaining information from the memory drum with a playback (read) head.

Playback Head - Same as "Read Head". See definition of "Read-Write Head" and page IV-18.

Plugboard - The interconnection panel of the IBM Model 523 Summary Punch.

Preamplifier - A linear amplifier used to increase the amplitude of signals from a read head or from the read winding of a read-write head.

Previous History Effect - The variation in the magnetic level of a drum channel, depending upon the information previously recorded in that channel and its residual effect.

Prime - The negation of a logical proposition. See page D-3.

Product Resistor - A resistor used with two or more diodes to form a logical product.

Program - A list of commands (set of instructions arranged in proper sequence by the programmer) which is entered into the computer to accomplish the solution of a certain problem, or the processing of a set of data.

Program Counter - The electronic circuitry within the computer, consisting of nine flip-flops ($N_1 - N_9$) and associated circuits, which selects the logical sequence of processes which the computer must go through in carrying out a command. The program counter determines which of a set of processes

the computer is performing at any given time. It is an internal control unit, and the operator has no programmed access to it.

Program Counter Number - The octal number referring to one state of the program counter (or to the associated block on the Flow Diagram).

Program Counter Sum - The logical sum of several program counter propositions.

Proposition - (1) In the 102-A, any point in a logical circuit which must have one of the two logical reference levels, either high (true, +125v) or low (false, +100v). (2) An expression in symbolic logic which can be verified as being true or false.

Pulse Counter - An electronic counter used to count the three pulse positions within each of the 14 octal digits in a word. The three pulse positions, corresponding to three bits used to define each octal digit, are identified as P_0 , P_1 , and P_2 . See typical Block Explanation Form, in Volume II, and also page IV-13, in this Volume.

Punch Brush - One of the contact brushes in an IBM unit which reads the holes in an IBM card.

Putaway - The computer operation of storing a command or number in a specified memory cell.

Radix - Same as "Base".

Radix Point - The index which separates the digits associated with negative powers from those associated with the zero and positive powers of the base of the number system in which a quantity of represented. For example, binary

point, octal point, decimal point.

Read - Same as "playback".

Read-Write Head - The electromagnetic element used to read information from the memory drum or to write information on the drum. A read-write head performs both functions for each of the 16 channels of the main memory section of the drum, while separate read and write heads are used for the recirculating registers, E, F, G, H, and J. There are read heads for the word channel and synchronizing pulse channel. The clock channels have read heads **which** are also used for recording the clocks. See page IV-18.

Recirculating Line - The delay-line registers (E, F, G, H, and J) circuitry is so designed as to couple the output and input so that the contents of the registers recirculates during specific blocks. Also known as "Recirculating Registers". See page IV-19.

Record (verb) - To place information on the magnetic drum with a write (record) head

Record Head - Same as "Write Head". See Read-Write Head.

Record Tube - A tube and associated circuitry used to supply current to the write or record heads.

Register - A cell or group of cells outside of the main memory, which are used in the performance of particular operations in any given program. In the 102-A Computer, the E, F, G, and H registers are one-word registers. The J (buffer) register contains eight words.

Relative Programming - A program-

ming technique whereby addresses assigned to commands do not refer to a particular cell but rather to the relative position with regard to some subsequently assigned arbitrary cell.

Rest - Same as "Idle".

Round-Off - To delete the least significant digit or digits from a number.

Routine - A sequence of operations which the computer may perform, or the sequence of commands which determine these operations. Examples are the "add routine", "print routine", etc. Also, used synonymously with "program".

Row - Referring to an IBM card, a horizontal arrangement of data.

R Row - The topmost row on an IBM card; the 12 row.

Sampling Clock - A series of 1.3-micro-second pulses synchronized with the clock and used in the recirculating register read circuits to eliminate the effects of the record transients induced in the read winding by pulses in the nearby record winding. See page IV-26.

Scale Factor - (1) The arbitrary factor which may be associated with a number in the computer to specify the position of the radix point so that the significant digits occupy specified positions in the word. (2) A command. See page III-9.

Sector - One of the peripheral elements into which a channel of the memory drum is divided. There are 64 sectors in each channel.

Sequencing (Power Supply) - The operation of the relays in the computer power supply in a carefully planned sequence so that d-c voltages will be turned on in such a way as not to injure the

logical diodes or alter the magnetic pattern.

Sequential Control - The manner of operation in which instructions or commands to the computer are set up in sequence and are fed consecutively to the computer during the solution of a problem.

Serial Transmission - A system of information transmission in which the characters of a word are transmitted in sequence over a single line, as contrasted with transmission (usually simultaneously) over separate lines in Parallel Transmission. The 102-A operations primarily are accomplished with serial transmission.

Shift - The operation of the computer which moves the bits in a word to the right or left.

Sign Convention - See pages III-1 & III-2 for the octal and decimal sign conventions, that is, the code signifying algebraic sign of numbers in the 102-A Computer.

Sign Digit - A character used to designate the algebraic sign (+ or -) of a number.

Sign Section - That section of a number-word consisting of the most significant six bits, positions 0_{12} and 0_{13} , and containing the numerical code for the algebraic sign of the number.

Significant Digits - A set of digits from consecutive positions in a number beginning with the most significant digit different from zero and ending with the least significant digit. The significance of a digit is greater when it occupies a position corresponding to a higher power of the base.

Skip - The movement of the program counter from one block to another which is not the next block in numerical order; possible in the 102-A only when the decision flip-flop (K) is true.

Start Proposition - A proposition which goes true when any significant Flexowriter key is struck during the "FILL" operation or when the "COMPUTE" button on the control panel is pushed.

Start Read - A special signal from the computer to the Flexowriter which starts the paper-tape reader of the Flexowriter.

Stick - The third type of operation permitted the program counter (count and skip are the other two) when the decision flip-flop (K) is true. During the stick operation no change is made in the program counter configuration until K is set false for counting to the next block. (Equivalent to skipping back to the same block.)

Stop Code - A coded signal on the Flexowriter paper tape which stops the tape. It has no effect on the computer.

Storage - Any device into which information can be introduced and then extracted at a later time: any of the registers or the main memory of the 102-A Computer; paper tapes of the Flexowriter; magnetic tapes of the CRC 126 Tape Units; cards of the IBM Units.

Subroutine - Part of a routine, usually prepared in advance to facilitate programming.

Sum Resistor - A resistor used with two or more diodes to form a logical sum.

Symbolic Logic - As applied to the 102-A, a Systematic technique for the design of bi-stable electronic circuitry.

Synchronizing Pulse - A pulse recorded at the P_2^0 position of all sectors of the word channel, (see page IV-23.)

Tab - A control key on the Flexowriter which, when pressed, performs two operations: (1) shift of information in the computer from the E (input) register to the main memory; and (2) accomplishes the normal tab operation of a typewriter.

Tape Unit or Tape Drive Mechanism-(TDM) - A Model CRC 126 Magnetic Tape Handling Unit which provides automatic auxiliary memory storage. See page I-2.

Transfer - The computer operation consisting of moving the contents of one cell or register to a second cell or register.

Transfer Command - A command which, conditionally or unconditionally, causes the next command word to be selected from a specified memory location.

Transfer of Control - Result of a transfer command operation in which the conditions for a decision are met.

Trouble - Location Problem - A test problem whose incorrect solution supplies information on the location of faulty equipment; used after a check or other problem has shown that a fault exists.

True - (1) State of a proposition when the voltage is at the higher of two logical reference levels; +125 volts in the 102-A. (2) State of a flip-flop which will make its associated output proposition true; also called its "one" or "on" state.

Tube Chassis - One of four chassis, placed two at each end of the 102-A Computer, with provision in each chassis for a large number of plug-in units (flip-flops, drivers, etc.). See Figure I-14.

Twelve (12) Row - See "R Row".

Upper Clamp - A diode used to fix the voltage swing of a given circuit point at the higher of two reference levels, namely +125 volts in the 102-A.

Unconditional Transfer - A command which always causes the following command to be taken from a specific non-sequential address.

Word - The basic data unit within the

computer; the content of any single cell. A word may be either a number or a command. See page III-1 for word structure.

Word Channel - A special channel used to identify the 64 sectors of the memory drum and whose output is called M_w . This channel is numbered octally 00 through 77 in the LSOD's of each of the three possible address positions and serves to indicate the time when any desired sector comes under a read or write head. Numbering M_w in special patterns facilitates minimum access programming.

Write - Same as "Record".

Write Relay - That relay in the computer which permits writing information on the magnetic tape of a Model 126 Tape Unit.

X Row - That row on an IBM card which appears immediately above the 0 (zero) row; the second row from the top of the card; the 11 row.

X Switches - A series of 10 test switches on the computer test panel, nine of which (N_1 through N_9) determine a particular block. When the tenth switch, marked X, is on, the computer will proceed with its normal operations through this specified block and then will skip back to rest. See page V-6.

X (Test) Proposition - The logical proposition generated by the X switches.

Y Switches - A series of 11 test switches on the computer test panel, nine of which (N_1 through N_9) determine a specific block by setting up its octal number in binary form. When a tenth switch, marked Y, is on, the computer will skip from rest to this particular block. The eleventh switch, K, sets the decision flip-flop to the desired condition (true or false) when entering the specified block. See page V-6.

Basic Systems and Arithmetic Processes

The 102-A Computer is designed to utilize three number systems: binary, octal, and decimal. All internal operations of the computer are binary (base 2). Input to, or output from, the computer may be in either octal (base 8) or decimal (base 10) form at the choice of the programmer, although utilization of decimal input-output requires conversion to binary within the computer.

BINARY NUMBER SYSTEM

Because the binary system has only two digits, 0 and 1, it is ideal for simple electronic representation. Any electric device with two stable states, such as a switch turned on or off, a flip-flop circuit, a magnetic surface saturated in either of two directions, a vacuum tube conducting or cut off, and many others, can be used to simulate binary numbers. All the operations in binary arithmetic can be represented in very simple form, using the two digits. Hence the binary system, expressed electronically, is used for all numbers, commands, and operations, within the 102-A Computer.

In the binary system, as in all other number systems, digits to the left of the radix point (in this case, the binary point) are coefficients of the zero power and consecutively increasing positive powers of the base; digits to the right of the radix point are coefficients of ascending negative powers of the base.

Example: Take the binary number 101101.01. The number to the left of the

binary point is $1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 8 + 4 + 1 = 45$ in decimal.

The number to the right of the binary point is $0 \times 2^{-1} + 1 \times 2^{-2} = 1/4 = 0.25$. Thus the binary number 101101.01 = 45.25 in decimal form. The rules for binary addition and multiplication are given in TABLES B-I and B-II.

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 10$ (zero and a carry)

TABLE B-I. Binary Addition

$0 \times 0 = 0$
$0 \times 1 = 0$
$1 \times 0 = 0$
$1 \times 1 = 1$

TABLE B-II. Binary Multiplication

Examples of Binary Arithmetic

In binary addition, a useful rule is that an even number of 1's gives a sum digit of 0; an odd number of 1's yields a sum digit of 1.

Adding numbers of the same sign:

	Decimal Check
+ 11001.01	25.25
+ 10101.11	21.75
+101111.00	47.00

Adding two numbers of opposite sign:

$$\begin{array}{r} + 11001 \\ - 10111 \end{array}$$

The method which the computer uses to perform this addition is to complement (see Glossary) the number with the negative sign, and then to add this true complement to the positive number. True complementing of a binary number consists of changing all 1's to 0's, and all 0's to 1's; and then adding 1 to the LSB position of the resulting number.

Thus the true complement of 10111 is 01001, and the above addition becomes

+ 11001	Decimal Check
+ 01001	+11001 = +25
<u>100010</u>	-10111 = -23
	<u>+ 2</u>

Answer = 00010

The procedure in this binary addition is to add the absolute values of the positive number and the complemented negative number. Then examine the bit in the overflow position (most significant bit) to determine the sign of the result. Whether this bit is 1 or 0, it is discarded. If it is 1, the sign of the answer is positive. If it is zero, the sign is negative, and the answer must be uncomplemented before it is correct.

An example of this latter case, where the negative number is larger than the positive number, and therefore the addition will yield a negative sum, is as follows:

$$\begin{array}{r} + 11001 \\ - 11110 \end{array}$$

Complementing the negative number gives 00010.

$$\begin{array}{r} \text{Adding} \quad 11001 \\ \quad 00010 \\ \hline 011011 \end{array}$$

However, in this sum the overflow digit is zero because the negative number was

larger than the positive number. This sum must be uncomplemented to obtain the correct answer. Uncomplementing (or recomplementing) involves exactly the same process as true complementing. Hence 11011 becomes 00101. Because the negative number was larger, the answer must have a negative sign. Thus:

$$-00101 = \text{Answer}$$

$$\begin{array}{rcl} \text{Decimal check:} & +11001 & = +25 \\ & -11110 & = -30 \\ & \hline & -00101 & = -5 \end{array}$$

Binary subtraction is a special case of addition, and merely consists of changing the sign of the subtrahend and then adding the latter to the minuend. For example:

Subtract 11010 from 110101 becomes
add +110101 minuend
-011010 subtrahend

As before, 011010 is complemented to obtain 100110.

$$\begin{array}{r} \text{Now, adding} \quad 110101 \\ \quad 100110 \\ \hline 1011011 \text{ Interpreted} \\ \text{as } +11011 \end{array}$$

$$\begin{array}{rcl} \text{Decimal check:} & & \\ \text{Subtract} & 110101 & = 53 \\ & 11010 & = 26 \\ & \hline & 11011 & = 27 \end{array}$$

Binary multiplication can be done directly as follows:

1110	Decimal check:
1011	14
<u>1110</u>	11
1110	<u>14</u>
0000	14
1110	<u>154</u>
<u>10011010</u>	

$$10011010 = 128 + 16 + 8 + 2 = 154$$

Another method of binary multiplication, which corresponds to the computer's process, consisting of placing the numbers in registers and then proceeding the same numbers as before, multiply 1011 by 1110.

	Work Reg.	Multiplier Reg.	Multiplicand Reg.	Description of Operation
(1)	0000	1110	1011	(1) Multiply Multiplicand by LSB in Multiplier (0, in this case) and add result to Work Register.
(2)	0000	0111	1011	(2) Having used LSB in Multiplier, shift Multiplier right one digit to eliminate this digit, and shift in LSB from Work Register in MSBD position of Multiplier. Shift Work Register right one digit, bringing 0 into MSBD position.
(3)	1011	0111	1011	(3) Repeat process of (1) above.
(4)	0101	1011	1011	(4) Repeat process of (2) above.
(5)	10000	1011	1011	(5) Repeat (1).
(6)	1000	0101	1011	(6) Repeat (2).
(7)	10011	0101	1011	(7) Repeat (1).
(8)	1001	1010	1011	(8) Shifting Work Register and Multiplier right one digit gets rid of final digit in original Multiplier. Answer is contained in Work Register and Multiplier Register.
Answer (as before) = 10011010				

Binary division can be performed directly as follows:

Divide 11011 by 1001

$$\begin{array}{r}
 11 \\
 1001 \overline{) 11011} \\
 \underline{1001} \\
 1001 \\
 \underline{1001} \\
 1001
 \end{array}$$

Decimal check $27/9 = 3$

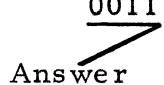
Quotient is $11_b = 3_d$

However, as above in multiplication, there is another method of division which corresponds more closely to the computer's process. Again registers are used and the procedure is as follows, using the same numbers employed above to divide directly.

That is, divide 11011 by 1001

	Dividend Registers	Divisor Register	Description of Operation
(1)	0001 1011	1001	(1) Place dividend in two 4-digit Registers.
(2)	1101 1001	1001	(2) Shift dividend left until the number in first Dividend Register is larger or equal to Divisor. Now Divisor will go into this number once, therefore a 1 is placed in the LSB in the second Dividend Register; this is the first quotient digit.

(3)	1101	1001	1001	(3) Subtract Divisor from number in first Dividend Register. This means complementing Divisor and adding the complement to Dividend.
(4)	10100	1001	1001	(4) Addition (or subtraction) is complete and overflow digit is discarded.
(5)	1001	0011	1001	(5) Repeat process in (2) above, and obtain second digit of answer.
(6)	1001	0011	1001	(6) Repeat (3).
(7)	10000	0011	1001	(7) Repeat (4), and obtain sum of zero, so division is complete. The answer is in the second Dividend Register, or 0011 as previously obtained by direct division.



It is to be noted that in the Model 102-A Computer divisions equal to or greater than 1 yield an overflow. To obtain a quotient that does not overflow, the division must be programmed with the dividend and divisor each expressed as less than 1. The quotient then, being less than 1, will not overflow. Thus, in the preceding example, 27 and 9 could be programmed as 0.27_d and 0.9_d or as 0.027_d and 0.09_d . The quotient in either case is 0.3_d --no overflow.

OCTAL NUMBER SYSTEM

TABLE B-III shows the relationship between binary, octal, and decimal numbers.

<u>Binary</u>	<u>Octal</u>	<u>Decimal</u>
000	0	0
001	1	1
010	2	2
011	3	3
100	4	4
101	5	5
110	6	6
111	7	7
1000	10	8
1001	11	9
1010	12	10
1011	13	11
1111	17	15
10000	20	16
10100	24	20
110010	62	50 etc.

TABLE B-III. Binary, Octal, and Decimal Equivalents

	1	2	3	4	5	6	7
1	2	3	4	5	6	7	10
2	3	4	5	6	7	10	11
3	4	5	6	7	10	11	12
4	5	6	7	10	11	12	13
5	6	7	10	11	12	13	14
6	7	10	11	12	13	14	15
7	10	11	12	13	14	15	16

TABLE B-IV. Octal Addition

	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	10	12	14	16
3	3	6	11	14	17	22	25
4	4	10	14	20	24	30	34
5	5	12	17	24	31	36	43
6	6	14	22	30	36	44	52
7	7	16	25	34	43	52	61

TABLE B-V. Octal Multiplication

Examples of octal arithmetic:

$$\begin{aligned}\text{Addition: } 752 &= 7 \times 8^2 + 5 \times 8^1 + 2 \times 8^0 = \\ &\underline{447} = 4 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 = \\ 1421 &= 1 \times 8^3 + 4 \times 8^2 + 2 \times 8^1 + 1 \times 8^0 =\end{aligned}$$

$$\begin{array}{r} \text{Subtraction: } 13201 \text{ (octal)} \\ - 4765 \\ \hline 6214 \end{array}$$

$$\begin{array}{r} \text{Multiplication: } 567 \text{ (octal)} \\ 734 \\ 2734 \\ 2145 \\ 5101 \\ \hline 534504 \end{array}$$

$$\begin{array}{r} \text{Division: } \\ 521 \overline{) 226565} \\ \underline{1763} \\ 3026 \\ \underline{2504} \\ 3225 \\ \underline{3225} \end{array}$$

$$\text{Answer: } 345_o = 229_d$$

Decimal check

$$490$$

$$\underline{295}$$

$$785$$

$$\begin{array}{r} 5761 \text{ (decimal)} \\ - 2549 \\ \hline 3212 \end{array}$$

$$\begin{array}{r} 375 \text{ (decimal)} \\ 476 \\ 2250 \\ 2625 \\ 1500 \\ \hline 178500_d = 534504_o \end{array}$$

Decimal check

$$\begin{array}{r} 229 \\ 337 \overline{) 77173} \\ \underline{674} \\ 977 \\ \underline{674} \\ 3033 \\ \underline{3033} \end{array}$$

CONVERSIONS BETWEEN NUMBER SYSTEMS

Binary-to-octal and octal-to-binary conversion can be handled by inspection once the values of the octal digits 0 through 7 in binary digits (see TABLE B-III) are familiar. Because $8 = 2^3$, each group of three binary digits, moving either right or left from the binary point, has a single octal equivalent. Thus, in converting from binary to octal, mark the binary number off in triads from the binary point, and read off the octal number. Octal to binary, convert each octal number to three binary digits. Examples:

Binary 1101110110.0111 = 001 101 110
110 . 011 100 = 1566.34 octal

Octal 7642.035 = 11111010001
0.000011101 binary

The rapidity and simplicity of this conversion back and forth between binary and octal numbers is a great convenience in using the 102-A Computer. Although all the internal operations of the machine are binary, the use of binary expressions for numbers and commands in the input-output devices would be laborious. Hence, all commands to the computer, and many numbers, are expressed octally. As previously mentioned, either octal or decimal numbers may be read into or written out of the 102-A, but decimal numbers must be converted to binary within the machine before the computer can operate upon them; and a subroutine for converting binary to decimal numbers is also necessary before decimal output is possible.

Binary-to-decimal and decimal-to-binary conversions are best handled by using octal numbers as the intermediate step, since octal-to-decimal (and the reverse) is a somewhat easier conversion.

Octal-to-decimal conversion is a

straightforward matter of multiplying each octal digit by the appropriate power of eight and adding the products. For example, octal $6735.24 = 6 \times 8^3 + 7 \times 8^2 + 3 \times 8^1 + 5 \times 8^0 + 2 \times 8^{-1} + 4 \times 8^{-2} = 3549.3125$ decimal.

Decimal-to-octal conversion is readily handled by using a table of powers of eight, and considering these powers as divisors of the decimal number. For example, in converting decimal 569.65 to octal, it is first divisible by 512 (8^3) which goes in once with remainder of 167.65.

This is divided by 64 (8^2) which goes twice, with remainder of 39.65; and the process is continued. Result:

decimal $679.65 = 1 \times 8^3 + 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0$
 $+ 5 \times 8^{-1} + 1 \times 8^{-2} + 4 \times 8^{-3} = 1247.514$ octal

Another technique for decimal-to-octal conversion consists of repeatedly dividing the decimal whole number by 8, so treating all numbers to the left of the decimal point. The remainders, read in reverse, give the octal number. Decimal numbers to the right of the decimal point are multiplied by 8, and the most significant digit of each product is removed and used as the next digit, in order of the octal fraction. Using the same number employed in the preceding example, decimal 679.65, the procedure is as follows:

8)679	Remainder	.65
8)84	7	<u>8</u>
8)10	4	5 <u>.20</u>
8)1	2	<u>8</u>
	1	1 <u>.60</u>
Answer: 679 _d = 1247 _o		<u>8</u>
		4 <u>.80</u>
.65 _d = .5146 _o		<u>8</u>
		6 <u>.40</u> etc.

As another example, convert 5978.42 decimal into the corresponding octal number.

8)5978	Remainder		.42	
8) <u>747</u>	2		<u>8</u>	
8) <u>93</u>	3	3	.36	
8) <u>11</u>	5		<u>8</u>	
8) <u>1</u>	3	2	.88	
	1		<u>8</u>	
		7	.04	etc.

Answer: $5978.42_d = 13532.327+_o$

Basic Symbolic Logic

Symbolic logic, as applied to the 102-A Computer, is a systematic technique for the design of bi-stable electronic circuitry. Thus it is a method for expressing the arithmetic of the binary number system in logical electronic form, as well as a method for describing logical procedures.

Boolean algebra is an algebraic system for manipulating symbolic logic. The basic operations of Boolean algebra can be expressed in terms of truth tables for various algebraic operations.

Negation or not is the change of a logical proposition from its true condition to its false condition, and vice versa. (See the Glossary for definitions of proposition, true, and false.) This can be accomplished physically by putting a signal into a driver; the output of which becomes the inverse of the input signal. TABLE C-I, the truth table for negation, is as follows; with 0 representing a false state, and 1 a true state; A is any proposition and A' is the negation of this proposition. In general, as applied to the 102-A, the conditions of true, a high reference voltage level (+125v), and binary 1 are synonymous, their counterparts, or negations, are false, a low reference voltage level (+100v), and binary 0.

A	A'
0	1
1	0

TABLE C-I. Truth Table for Negation

Another truth table may readily be constructed for making the logical sum of any propositions, which corresponds electronically to mixing and is often called the or proposition. It is so called because in the process of logical addition of two propositions A and B, the sum is true if A or B is true, or both are true. That is; if at least one of the propositions which are combined to form a logical sum is true, the sum itself is true. TABLE C-II is a truth table, using two propositions, A and B, to illustrate the process of obtaining the logical sum:

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

TABLE C-II. Truth Table for Logical Sum

A similar truth table may be constructed for the logical product, which corresponds electronically to gating, and is known as the and proposition. This name is suitable because if the logical product of two propositions A and B, for example, is to be true, both A and B must be true. TABLE C-III illustrates this point.

A	B	AB
0	0	0
1	0	0
0	1	0
1	1	1

TABLE C-III. Truth Table for Logical Product

LIST OF IDENTITIES IN BOOLEAN ALGEBRA

The following list of identities, using A, B, and C as logical propositions, should prove useful in simplifying logical expressions and verifying the relations between them.

Sums: $A + B = B + A$

$$A + A = A$$

$$(A + B) + C = A + (B + C)$$

$$A + 1 = 1$$

$$0 + A = A$$

$$A + A' = 1$$

Products: $AA = A$

$$AB = BA$$

$$A(BC) = (AB)C$$

$$1 \cdot A = A$$

$$AA' = 0$$

$$(AB)' = A' + B'$$

$$(A + B)' = A'B'$$

$$0 \cdot A = 0$$

Note: $(AB)' \neq A'B'$

$$(A+B)' \neq A'+B'$$

It is possible to show that a typical identity, $(A+B)(A+C) = A+BC$, is true by means of TABLE C-IV.

As the values of the functions in the two right-hand columns are equal for all possible permutations of values of the

individual propositions, the two functions themselves are equal: or $(A+B)(A+C) = A+BC$.

A	B	C	A+B	A+C	(A+B)(A+C)	A+BC
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	0	0	0
1	0	0	1	1	1	1
1	0	1	1	1	1	1
0	1	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

TABLE C-IV. Identity Truth Table

It is possible to prove the validity of any of the identities in the preceding list by the same technique. It is worthwhile for the reader to prove for himself, by means of truth tables, several of these identities.

The following is an example of Boolean algebra showing the use of logical sums and logical products in combined expressions:

Simplify the following logical sum by factoring and using logical identities:

$$ABC' + A'BC' + A'BB'C + A'B'C + A'B'C' = (A+A')BC' + BB'(A'C) + A'B'(C + C') = BC' + A'B'$$

In APPENDIX D there are several examples to show how this symbolic logic is used in the design of the computer circuitry.

Logical Circuit Development

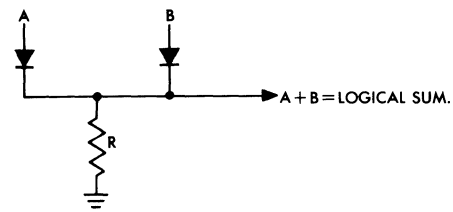
The Model CRC 102-A computing circuits have been designed by means of two-valued symbolic logic. The logic is handled in mathematical form according to a set of rules called Boolean algebra. A proposition in the computer is analogous to a proposition in semantics: it can be in one of two states, true or false. In the computer, these two states are represented by two voltage values. Thus, the output of any logical network or circuit constrained to these logical voltage levels becomes a proposition.

The two voltage levels selected to utilize this algebra in the computer are +125 and +100 volts. This high-low voltage condition is used to represent 1 or 0 (the two numbers used in binary arithmetic) or, with reference to a proposition, true or false. The logical bistable element used in the computer is the vacuum tube flip-flop; its two states are the extremes of this 25-volt range between the logical voltages. Also, all signals in the logical networks are constrained to one of these two voltage values.

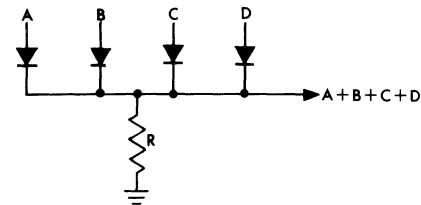
By defining certain logical operations, propositions are combined in various ways to achieve certain desired electronic results. The operations used are "and" (logical product, gating, or coincidence), "or" (logical sum or mixing), and "negation" (prime, not, or inversion). The basic electronic elements used to perform these operations are diode nets, flip-flops, drivers or inverter tubes, and cathode-followers.

LOGICAL SUM, PRODUCT AND NEGATION IN ELECTRONIC FORM

The sum of diodes in the formation of a logical sum is illustrated below. To express the logical sum of electronic propositions (voltages) A and B, the schematic is



Or, to obtain the logical sum of four propositions, $A + B + C + D$, the circuit is



These schematics indicate that if A is at the true level (+125v), then the sum $A + B$ (A added logically to any number of other propositions) will be at approximately the same level. This can be proved by examining the four possible cases of two propositions, A and B added logically (see Figure D-1).

In cases 1 and 4 of Figure D-1, the forward resistances (R_f) of the diodes have been substituted for the symbols of the diodes. R , the resistor to ground, is much larger than R_f . Typical values

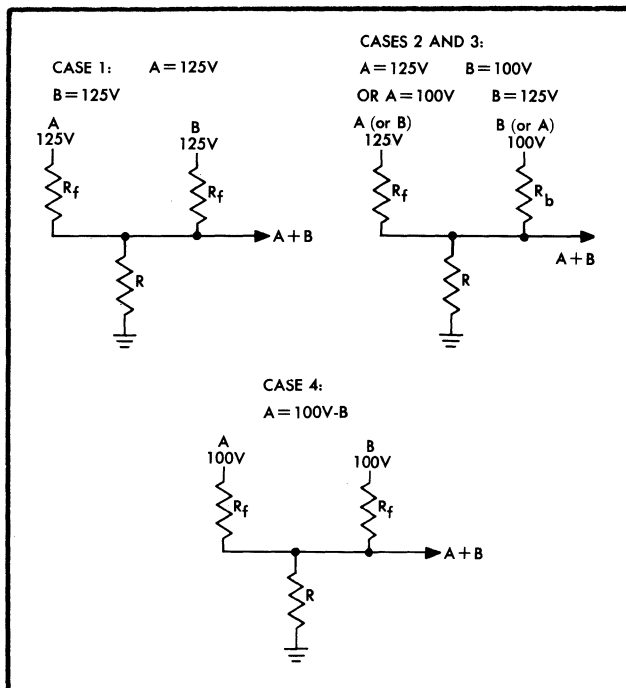
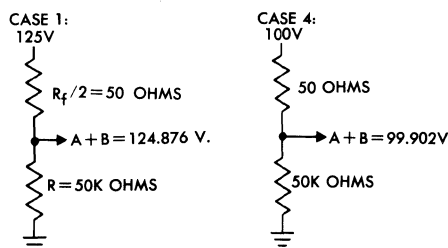


Figure D-1. Sum Propositions

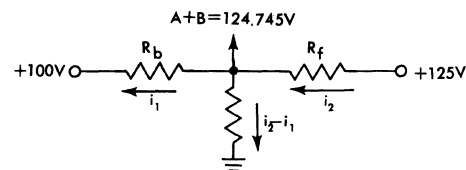
might be $R_f = 100$ ohms and $R = 50k$ ohms, as noted below.



These voltages for the logical sum, $A + B$, are obtained by considering the circuits as voltage dividers.

Cases 2 and 3 of Figure D-1, which are symmetrical, present a different problem. As before in the case of the true input (125v), which may be either A or B, the diode may be replaced by its forward resistance (R_f); however, the other diode, connected to the false level of 100v, must be represented by its back resistance R_b . This can be seen by inserting only the diode representing R_f

accordingly, as demonstrated in Case 1 above, the voltage at the cathode will be very nearly 125v. Now if the other diode (connected to the 100v supply) is inserted, forward current cannot flow through it because its cathode is at a higher potential than its plate. Hence its back resistance must be considered, and Cases 2 and 3 can be represented by the following equivalent circuit



The equations for this circuit are:

$$(i_2 - i_1)R - i_1 R_b = 100$$

$$(i_2 - i_1)R + i_2 R_f = 125$$

Solving for i_2 produces $i_2 = \frac{125R_b + 25R}{R_f R_b + R_f R + R_b R}$

where

$$R_f = 100 \text{ ohms}$$

$$R_b = 500K \text{ ohms}$$

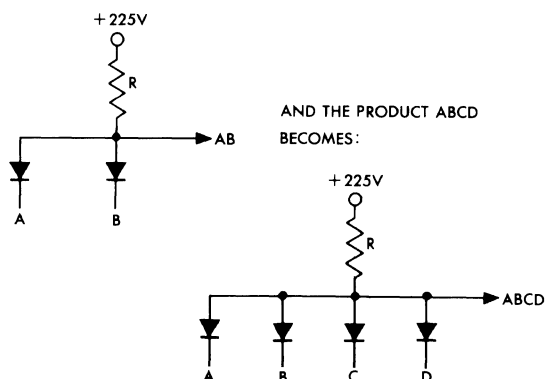
$$R = 50K \text{ ohms}$$

Thus, $i_2 = 2.55$ ma and the drop across $R_f = 0.255$ v.

Again the voltage at the output point of the logical sum is very nearly 125v; in Cases 2 and 3, it is 124.7v. Thus, the original statement has been proved that if proposition A is at the true level of 125v, then the logical sum $A + B$ will be at very nearly the same level of 125v.

It is worth noting that the sum resistor (R) must be much greater in value than the forward resistance of a diode. If R were too low, it would be impossible to get the sum up to the proper level (approximately 125v or 100v); and also, there would be excessive current flow through the diode.

The use of diodes in the formation of a logical product is illustrated below:



Using circuits similar to those for the various cases of logical addition, it can be shown that if both A and B are at the level of 125v, then the logical product AB will be at approximately 125v. Conversely, the product will be false (100v) if either or both A and B are false.

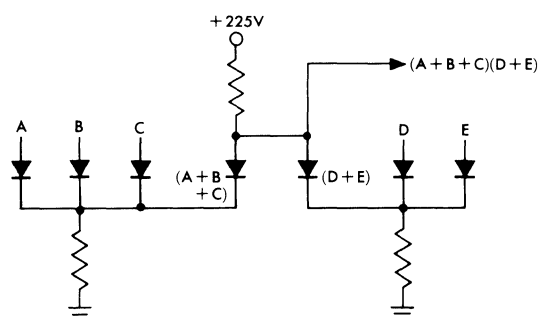
To express the logical sum and logical product of two propositions, A and B, in terms of voltages, the truth table shown in TABLE D-I may be constructed.

A	B	A+B	AB
100	100	100	100
100	125	125	100
125	100	125	100
125	125	125	125

TABLE D-I. Sum and Product Truth Tables Expressed in Voltages

Note that TABLE D-I is a composite of the previous TABLES C-II and C-III for logical sum and logical product, respectively, merely substituting 125 for 1 and 100 for 0.

In the following circuit logical sum and product circuits are combined to form a single logical network, typical of those used in the computer. Here, the product of two sums $(A+B+C)(D+E)$ is obtained in electronic form.



Negation can be expressed electronically in several ways. One example is placing a positive voltage change on the grid of a driver (direct-coupled amplifier) tube. The resulting voltage change on the plate of the driver is inverted, the negation of the input. Thus if the input is at the higher of two possible levels, the output will be at the lower of two possible levels; and vice versa. Another example is the flip-flop, since one of its two outputs is the prime (negation) of the other.

Programs

RELIABILITY

PART I

-02000-26210021000000*	2001-35200020042000
2002-34200520002000*	2003-11210021001700
2004-00000000000001*	2005-26210021002000
2000f- Compute *	
0000-34300021001310 (main rou-	0001-05300030000022
0002-35210021002003 time starts	0003-35200021002006
0004-35200320062003 here.)	0005-35200020062006
0006-34200220060004	0007-36200420032002
0010-34200221000053	0011-05300030000021
0012-35210021002002	0013-35210020012004
0014-26200420042006	0015-35200220062002
0016-35200120042004	0017-34200320040014
0020-34300021000031	0021-00000000000001
0022-000000000002000	0023-000000000001000
0024-00000001777000	0025-00000252125400
0026-00000000000003	0027-00000000000002
0030-00000000000001	0031-36200520022001
0032-34200121000056	0033-05300030000071
0034-35210021002002	0035-35200121002004
0036-26200420042006	0037-26200420062007
0040-35200220072002	0041-35200420012004
0042-34200320040036	0043-36200520022001
0044-34200121000061	0045-05300030000072
0046-36200220062002	0047-30200620042006
0050-34200621000046	0051-34200221000064
0052-34300021000101	0053-35210000712001
0054-21200121000003	0055-34300021000011
0056-35007400302000	0057-21200021000003
0060-34300021000033	0061-35210000262000
0062-21200021000003	0063-34300021000045
0064-35002600742001	0065-21200121000002
0066-34300021000052	0067-21000000770400
0070-34300021000001	0071-00000000000001
0072-00777777777777	0073-00000000000600
0074-00000000000001	0075-00050224110000

*This is an Erase Memory routine. After memory is erased, command in 2003 automatically starts tape reader to fill main program.

0076-00000000000001
0100-00000000000001
0102-36210020022002
0104-36200220062002
0106-34200621000104
0110-05300030000072
0112-36200220062002
0114-34200621000112
0116-36013220032004
0120-05300030000072
0122-36200220062002
0124-30200620042006
0126-27200201372003
0130-34200421000143
0132-00777777777777
0134-21200121000002
0136-00000000000005
0140-35003001362001
0142-34300021000120
0144-21200121000002
0146-05300030000161
0150-27200020072001
0152-34200121000147
0154-34200321000162
0156-00000000000000
0060-34300021000033
0062-21200021000003
0064-35002600742001
0066-34300021000052
0070-34300021000001
0072-00777777777777
0074-00000000000001
0076-00000000000001

0100-00000000000001
0102-36210020022002
0104-36200220062002
0106-34200621000104
0110-05300030000072
0112-36200220062002
0114-34200621000112
0116-36013220032004
0120-05300030000072
0122-36200220062002
0124-30200620042006
0126-27200201372003
0130-34200421000143
0132-00777777777777
0134-21200121000002
0136-00000000000005
0140-35003001362001

0077-02000000000001
0101-05300030000072
0103-36210020062006
0105-30200620042006
0107-34200221000133
0111-36210020022002
0113-27200620042006
0115-27200201572003
0117-34200421000140
0121-36210020062006
0123-37210030000124
0125-34200621000122
0127-36013220032004
0131-34300021000146
0133-35013621002001
0135-34300021000110
0137-02000000000001
0141-21200121000002
0143-36013601572001
0145-34300021000131
0147-31210020012007
0151-30200120052001
0153-35016020072003
0155-34300021000166
0157-02000000000002
0061-35210000262000
0063-34300021000045
0065-21200121000002
0067-21000000770400
0071-00000000000001
0073-00000000000060
0075-00050224110000
0077-02000000000001

0101-05300030000072
0103-36210020062006
0105-30200620042006
0107-34200221000133
0111-36210020022002
0113-27200620042006
0115-27200201572003
0117-34200421000140
0121-36210020062006
0123-37210030000124
0125-34200621000122
0127-36013220032004
0131-34300021000146
0133-35013621002001
0135-34300021000110
0137-02000000000001
0141-21200121000002

0142-34300021000120	0143-36013601572001
0144-21200121000002	0145-34300021000131
0146-05300030000161	0147-31210020012007
0150-27200020072001	0151-30200120052001
0152-34200121000147	0153-35016020072003
0154-34200321000162	0155-34300021000166
0156-00000000000000	0157-02000000000002
0160-00000000000043	0161-00777777777777
0162-35002601362006	0163-21200621000003
0164-34300021000055	0165-02000000000001
0166-31210001562007	0167-35200701752006
0170-34200621000172	0171-34300021000177
0172-35017621002006	0173-21200621000003
0174-34300021000171	0175-00000000000100
0176-00000000000011	0177-05300030000221
0200-35210020012004	0201-35210020032002
0202-25200220012000	0203-35200420002004
0204-25200320022002	0205-36200500212005
0206-34200521000202	0207-36022220042003
0210-34200321000226	0211-35045721000212
0212-35047421000213	0213-35210021000213
0214-35210021000212	0215-05300030000230
0216-35210020002001	0217-34300021000233
0220-00000000000001	0221-00400000000000
0222-00223305225450	0223-02600000000000
0224-00000000000000	0225-00000000000020
0226-21200321000002	0227-34300021000211
0230-00000100000000	0231-00000000000000
0232-00000700000000	0233-23200120022003
0234-35200120002001	0235-35023300710233
0236-34041220010233	0237-35200420032003
0240-35200520032004	0241-35200620042005
0242-37210030000242	0243-35200720052006
0244-37210030000245	0245-36054120062007
0246-35055121000233	0247-34200721000251
0250-34300021000254	0251-36017601572006
0252-21200621000002	0253-34300021000250
0254-05300030000273	0255-24200720032000
0256-26200320012001	0257-35200120002001
0260-37200130000271	0261-36200720022006
0262-34200621000266	0263-35200720052007
0264-34200320070255	0265-34300021000300
0266-35017600262000	0267-21200021000003
0270-34300021000263	0271-35200200212002
0272-34300021000261	0273-00700000000000
0274-00000000000000	0275-00000100000000
0276-00777777777777	0277-00000100000000
0300-05300030000327	0301-32007203372100
0302-32016103242001	0303-32007203532002
0304-32016103652003	0305-32007203142004
0306-32016103262005	0307-32007203552006

0310-32016103672107	0311-21200003360010
0312-05300030000321	0313-34300021000340
0314-00030504040502	0315-00101010101010
0316-00202020202020	0317-00404040100301
0320-00010101010101	0321-00534312474753
0322-00233313113102	0323-00504363734371
0324-00604070707040	0325-00227677527476
0326-00406050603040	0327-00747573727274
0330-00671707573737	0331-00040101000401
0332-00203224400546	0333-00060707050400
0334-00201115001143	0335-00050304040606
0336-10000000000000	0337-00343675327445
0340-32035403172000	0341-32031603612001
0342-32031503602002	0343-32035703222003
0344-32037103342004	0345-32032003632005
0346-32033203752006	0347-32033003732007
0350-21200003360010	0351-34300021001231
0352-00000020000000	0353-00070505000204
0354-00404040404040	0355-00020105060701
0356-00000000000001	0357-00040404040404
0360-03220312103003	0361-00040461070024
0362-03456464674164	0363-00424431122140
0364-00747575757475	0365-00707040406040
0366-00273767677737	0367-00636075456364
0370-00020514273436	0371-00020202020202
0372-00504070606060	0373-00707070707070
0374-00404040604070	0375-00070707070707
0376-00307040507050	0377-34300021000425
0457-35047421000213	0474-35210021000213
0412-00000600000000	
0541-00111111111111	0551-23200120022003
0001f-	

PART II (00400-1677) and PART III (1310-1755)

0400-34164311061224	0401-34300021001621
0402-34300021001224	0403-34300021001527
0404-02000000000007	0405-00000000000000
0406-33045115141773	0407-34300021001621
0410-34300021001773	0411-00000000000001
0412-00000600000000	0413-00000000000001
0414-33105714220501	0415-34300021001621
0416-34300021000501	0417-33146206251604
0420-34300021001621	0421-34300021001604
0422-33016511300607	0423-34300021001621
0424-34300021000607	0425-37147030000406
0426-34147121001232	0427-35013221001471
0430-34300021000651	0431-02000000000001
0432-00000000000000	0433-02000000000001
0434-00777777777777	0435-05300030000716
0436-35200620002000	0437-35200220042002

0440-35200220002000	0441-34200021000447
0442-33110507501327	0443-34300021001621
0444-34300021001327	0445-21071504310010
0446-34300021000435	0447-21071604310010
0450-34300021000442	0451-00000000000002
0452-00000000000000	0453-02000000000001
0454-00000000000000	0455-00000000000000
0456-00000000000000	0457-35047421000213
0460-00000000000000	0461-04300030001030
0462-35210021001037	0463-35210021000452
0464-35210021000454	0465-34300021001354
0466-00000000000000	0467-33043214750654
0470-34300021001621	0471-34300021000654
0472-00000000000001	0473-00000000000000
0474-35210021000213	0475-34154006031021
0476-34300021001621	0477-34300021001021
0500-02777777777777	0501-33134411071616
0502-33064511101616	0503-33134604111616
0504-33074715120571	0505-34300021001621
0506-34300021000571	0507-21107505620001
0510-21103605620001	0511-34300021000531
0512-33105515200377	0513-34300021001621
0514-34300021000377	0515-00000000000003
0516-02000000000001	0517-05300030001065
0520-35200520072005	0521-35200620002006
0522-35200620052005	0523-34200521000605
0524-05300030001073	0525-35200320052003
0526-35200420062006	0527-35200320062003
0530-34200321000612	0531-35102710312000
0532-35103510372001	0533-35200020012000
0534-34200021000617	0535-34010013431561
0536-34300021001621	0537-34300021001561
0540-02000000000002	0541-00111111111111
0542-00000000000001	0543-00000000000001
0544-02777777777777	0545-33111004531332
0546-34300021001621	0547-34300021001332
0550-00000000000000	0551-23200120022003
0552-05300030000452	0553-35200220042002
0554-35200620002006	0555-35200220062002
0556-34200221001454	0557-33112113651544
0560-34300021001621	0561-34300021001544
0562-02000000000001	0563-00000000000001
0564-33002704720751	0565-34300021001621
0566-34300021000751	0567-00000000000001
0570-02000000000023	0571-26043413771065
0572-25133505001027	0573-26123611011067
0574-25153711021031	0575-24124011031073
0576-23134104041035	0577-24064210051075
0600-23104315061037	0601-34300021000517
0602-02525252525252	0603-00000000000000

0604-02000000000001	0605-21106500770001
0606-34300021001463	0607-33135216151574
0610-34300021001621	0611-34300021001574
0612-21107300770001	0613-34300021001532
0614-33105714221601	0615-34300021001621
0616-34300021001601	0617-21102704310001
0620-21103104310001	0621-21103513440001
0622-21103704310001	0623-34300021000535
0624-007777777777776	0625-00000000000000
0626-35210021002100	0627-34300021001621
0630-34300021000662	0631-34200421001412
0632-35210021000753	0633-32027612410753
0634-35123721000755	0635-32050015430755
0636-35050221000757	0637-32110213450757
0640-35054421000761	0641-34300021000741
0642-020000000000006	0643-34300021000467
0644-000000000000000	0645-020000000000001
0646-34300021001672	0647-000000000000000
0650-35210021001471	0651-34151410570475
0652-34300021001621	0653-34300021000475
0654-34111705621616	0655-34022005631616
0656-33002112641616	0657-33112213651044
0660-34300021001621	0661-34300021001044
0662-35210021001035	0663-34300021001700
0664-000000000000003	0665-100000000000001
0666-000000000000001	0667-020000000000043
0670-000000000000002	0671-000000000000004
0672-020000000000001	0673-000000000000000
0674-35013713020716	0675-35154013030715
0676-35104106040720	0677-35054211050717
0700-36054313060722	0701-36134411070721
0702-36064515100724	0703-36134611110723
0704-05300030000715	0705-35200520072005
0706-35200120032001	0707-35200520012005
0710-34200521000445	0711-34300021000435
0712-000000000000000	0713-000000000000000
0714-000000000000000	0715-000000000000000
0716-000000000000000	0717-000000000000000
0720-000000000000000	0721-000000000000000
0722-000000000000000	0723-000000000000000
0724-000000000000000	0725-000000020000000
0726-000000000000000	0727-000000000000000
0730-000000000000000	0731-33107316031616
0732-33164410341616	0733-33107516051616
0734-33164610361616	0735-33107716071616
0736-33165010301616	0737-33107116111616
0740-34300021001271	0741-32110410470761
0742-05300030000753	0743-35200320052003
0744-36200720012007	0745-35200320072003
0746-34300021001254	0747-000000000000001

0750-00000000000000	0751-34300021000426
0752-00000000000000	0753-00000000000000
0754-01000000000001	0755-00000000000000
0756-00000000000000	0757-00000000000000
0760-00000000000000	0761-00000000000000
0762-00000000000000	0763-00000000000000
0764-00000000000000	0765-00000000000000
0766-00000000000000	0767-00000000000000
0770-00000000000000	0771-00000000000000
0772-00000000000000	0773-00000000000000
0774-00000000000000	0775-00000000000000
0776-00000000000000	0777-00000000000000
1000-00000000000000	1001-00000000000000
1002-00000000000000	1003-00000000000000
1004-00000000000000	1005-00000000000007
1006-00000000000000	1007-00000000000002
1010-37135330001616	1011-37075430001472
1012-34300021001621	1013-34300021001472
1014-00000000000003	1015-00000000000000
1016-34016106241042	1017-34300021001621
1020-34300021001042	1021-34066400271245
1022-34300021001621	1023-34300021001245
1024-03000000000001	1025-00000000000000
1026-00000000000000	1027-00000000000000
1030-00000000000000	1031-00000000000000
1032-00000000000000	1033-00000000000000
1034-00000000000000	1035-00000000000000
1036-00000000000000	1037-00000000000000
1040-00000000000000	1041-00000000000001
1042-34300021001366	1043-02000000000006
1044-34110716521570	1045-34300021001621
1046-34300021001570	1047-00000000000000
1050-00000000000000	1051-00000000000000
1052-34051513601076	1053-34300021001621
1054-34300021001076	1055-00000000000000
1056-00000000000000	1057-00000000000000
1060-33142304660545	1061-34300021001621
1062-34300021000545	1063-00000000000000
1064-00000000000000	1065-00000000000000
1066-00000000000000	1067-00000000000000
1070-00000000000000	1071-00000000000000
1072-00000000000000	1073-00000000000000
1074-00000000000000	1075-00000000000000
1076-34300021001522	1077-00000000000000
1100-02000000000001	1101-00777777777777
1102-02777777777777	1103-00000000000007
1104-00777777777777	1105-00000000000001
1106-00000000000000	1107-00000000000001
1110-00000000000000	1111-02000000000001
1112-34105502201616	1113-33035611210400

1114-34300021001621	1115-34300021000400
1116-00000000000000	1117-02000000000001
1120-00000000000000	1121-00000000000000
1122-02000000000001	1123-00000000000000
1124-00000000000000	1125-00000000000000
1126-02020000000000	1127-00000000000000
1130-02000000000002	1131-00000000000001
1132-00000000000000	1133-00000000000000
1134-00000000000000	1135-00000000000000
1136-00000000000000	1137-00000000000000
1140-00000000000000	1141-00000000000000
1142-00000000000000	1143-00000000000000
1144-00000000000000	1145-00000000000000
1146-00000000000000	1147-00000000000000
1150-00000000000000	1151-00000000000000
1152-00000000000000	1153-00000000000000
1154-00000000000000	1155-00000000000000
1156-00000000000000	1157-00000000000000
1160-00000000000000	1161-00000000000000
1162-00000000000000	1163-00000000000000
1164-00000000000000	1165-00000000000000
1166-00000000000000	1167-00000000000000
1170-00000000000000	1171-00000000000000
1172-00000000000000	1173-00000000000000
1174-00000000000000	1175-00000000000000
1176-00000000000000	1177-00000000000000
1200-00000000000000	1201-00000000000000
1202-00000000000000	1203-00000000000000
1204-00000000000000	1205-00000000000000
1206-00000000000000	1207-00000000000000
1210-00000000000000	1211-00000000000000
1212-00000000000000	1213-00000000000000
1214-00000000000000	1215-00000000000000
1216-00000000000000	1217-00000000000000
1220-00000000000000	1221-00000000000000
1222-17204030001433	1223-34300021001436
1224-34056704320403	1225-34300021001621
1226-34300021000403	1227-00000000100000
1230-01042104210421	1231-17201030000650
1232-17202030001132	1233-17204030001433
1234-34300021001436	1235-02000000000002
1236-02777777777777	1237-02252525252525
1240-00000000000006	1241-00777777777777
1242-37130530001523	1243-34300021001621
1244-34300021001523	1245-34111013531616
1246-34041116541616	1247-34151210551616
1250-34041301560674	1251-34300021001621
1252-34300021000674	1253-00000020000000
1254-34200021001662	1255-05300030001642
1256-34002113641501	1257-34300021001621

1260-34300021001501	1261-37102430001242
1262-34300021001621	1263-34300021001242
1264-00000000000001	1265-02735673567356
1266-34113102740512	1267-34300021001621
1270-34300021000512	1271-33164210321616
1272-33160310731616	1273-33103416441616
1274-33160510751616	1275-33103616461616
1276-33160710771616	1277-34104113040422
1300-34300021001621	1301-34300021000422
1302-00000000000001	1303-02000000000001
1304-00000000000000	1305-01000000000001
1306-00000000000001	1307-00000000000015
1310-05210021002100	1311-04300030000712
1312-04300030001401	1313-04300030000715
1314-04300030001476	1315-04300030001064
1316-04300030001065	1317-04300030000716
1320-34300021000461	1321-34126400281616
1322-34016500301616	1323-34176602311347
1324-34300021001621	1325-34300021001347
1326-02000000000001	1327-33067212350414
1330-34300021001621	1331-34300021000414
1332-33147505400417	1333-34300021001621
1334-34300021000417	1335-00777777777777
1336-02000000000001	1337-22300030001340
1340-00000000000000	1341-00000000000006
1342-00000000000000	1343-00000000000000
1344-02000000000001	1345-00252525252525
1346-00000000000001	1347-33161210551534
1350-34300021001621	1351-34300021001534
1352-00000000000001	1353-00000000000001
1354-35210021000456	1355-35210021000460
1356-35210021001471	1357-34300021000626
1360-00000000000002	1361-21106713360001
1362-21103013360001	1363-34300021000524
1364-00000000000000	1365-02000000000002
1366-34113102741112	1367-34300021001621
1370-34300021001112	1371-00000000000000
1372-00000000000024	1373-00000000000000
1374-00000000000000	1375-01000000000001
1376-00000000000000	1377-00777777777777
1400-00000000000000	1401-00000000000000
1402-00000000000000	1403-00000000000000
1404-00000000000000	1405-00000000000000
1406-00000000000000	1407-00000000000000
1410-00000000000015	1411-00000000000000
1412-21106504310002	1413-21102613360002
1414-34300021000632	1415-00000000000000
1416-34016106241542	1417-34300021001621
1420-34300021001542	1421-00000000000001
1422-02000000000002	1423-00000000000001

1424-02400000000000	1425-34067004331451
1426-34300021001621	1427-34300021001451
1430-33047313361515	1431-34300021001621
1432-34300021001515	1433-11300030000762
1434-21076206650010	1435-21077213750013
1436-34300021001577	1437-00000000000000
1440-00000000000001	1441-00000000000000
1442-00000000000000	1443-00000000000000
1444-00000000000000	1445-00000000000000
1446-00000000000000	1447-00000000000000
1450-00000000000000	1451-34101401571675
1452-34300021001621	1453-34300021001675
1454-21045204310001	1455-21045404310001
1456-21045604310001	1457-21046004310001
1460-34300021000557	1461-02000000000002
1462-00000000000001	1463-21102614750001
1464-34300021001361	1465-33003015731052
1466-34300021001621	1467-34300021001052
1470-01000000000001	1471-00000000000000
1472-34133515001016	1473-34300021001621
1474-34300021001016	1475-02000000000001
1476-00000000000000	1477-00000000000000
1500-00777777777776	1501-04300021001030
1502-33103016501616	1503-33161110711616
1504-33103216421616	1505-34300021000731
1506-02000000000007	1507-00000000000000
1510-02000000000001	1511-00000000000000
1512-00000000000000	1513-00000000000000
1514-00000000000001	1515-34136014231541
1516-34300021001621	1517-34300021001541
1520-02000000000001	1521-00000000000000
1522-34300021000646	1523-34066602311010
1524-34300021001621	1525-34300021001010
1526-00000000000000	1527-34177212341553
1530-34300021001621	1531-34300021001553
1532-21103415400001	1533-34300021000507
1534-34177713421060	1535-34300021001621
1536-34300021001060	1537-02777777777777
1540-02000000000001	1541-34300021001465
1542-34300021001266	1543-02525252525252
1544-31130703521065	1545-31141012531066
1546-35106510662004	1547-35102610272003
1550-37200330000631	1551-34300021001621
1552-34300021000631	1553-33051617611616
1554-34111705621616	1555-33022005631616
1556-33002113640643	1557-34300021001621
1560-34300021000643	1561-30142406670452
1562-27072505700454	1563-30112606710456
1564-27122713720460	1565-33003004730552
1566-34300021001621	1567-34300021000552
1570-34043313760614	1571-34300021001621

1572-34300021000614	1573-00000000000000
1574-33013716021261	1575-34300021001621
1576-34300021001261	1577-17210030000000
1600-34300021001337	1601-34300021001425
1602-02000000000002	1603-02000000000001
1604-34300021001430	1605-02000000000001
1606-35000021002001	1607-02000000000001
1610-00000000000000	1611-02000000000001
1612-00000000000001	1613-00000000000000
1614-00777700000000	1615-00000000000000
1616-35300021002000	1617-36200016742007
1620-34300021001623	1621-35300021002000
1622-36200016732007	1623-32200716141624
1624-21000011000001	1625-35166116241626
1626-35000021002001	1627-32200116141630
1630-21000011000001	1631-30200116552002
1632-32200216141633	1633-21000011000001
1634-30200216552003	1635-32200316141636
1636-21000011000001	1637-30200016532004
1640-32200416601641	1641-34300021000000
1642-00000000000001	1643-00000000000001
1644-00000000000001	1645-21000011000001
1646-00000000000001	1647-00000000000000
1650-00000000000001	1651-00000000000000
1652-00000000000000	1653-020000000000030
1654-00000000000001	1655-000000000000014
1656-00000000000000	1657-00000000000000
1660-00000000007777	1661-35000010002000
1662-21075313260001	1663-21075513260001
1664-21075113260001	1665-21076113260001
1666-34300021001255	1667-00000000000000
1670-00000000000000	1671-00000000000000
1672-34300021001416	1673-00000200000000
1674-00000100000000	1675-34144006031321
1676-34300021001621	1677-34300021001321
1700-35210021001776	1701-35176521001703
1702-35210021001774	1703-05300030000000
1704-35200017742000	1705-37210021001706
1706-35200120002000	1707-37210021001710
1710-35200320002003	1711-37210021001712
1712-35200520032000	1713-37210021001714
1714-35200420002000	1715-37210021001716
1716-35200220002001	1717-37210021001720
1720-35200620012002	1721-37210021001722
1722-35200720021774	1723-37210021001724
1724-35170317601703	1725-34176717031703
1726-34177421001762	1727-35210021001774
1730-35177521001731	1731-05300030000000
1732-27200517562005	1733-35177620052005
1734-27200017562000	1735-35200020052005
1736-27200117562001	1737-35200520012005

1740-27200317562003	1741-35200520032001
1742-27200617562006	1743-35200620012001
1744-27200717562007	1745-35200120072005
1746-27200217562002	1747-35200220052007
1750-27200417562004	1751-35200420071776
1752-35173117571731	1753-34176417311731
1754-34177621001770	1755-34300021000001
1756-02000000000017	1757-00000000000010
1760-00000000000010	1761-00000000000001
1762-21177400770001	1763-34300021001727
1764-05300030001764	1765-05300030000000
1766-00000000000001	1767-05300030001764
1770-21177600770001	1771-34300021001755
1772-00000000000003	1773-34300021000564
1774-00000000000000	1775-05300030000000
1776-00000000000000	1777-02000000000001

Command Test

0000-

17201021000007	21020302650002	21020302660002	21020502670002
21021102070002	21021302100005	17210021000001	37022021000350
37022121000015	17204021000011	21001002650001	21022102650001
17210021000010	34022202230355	34022302220023	17204021000017
21001602650001	21022202650002	17210021000016	33022302220362
33022202230031	17204021000025	21002402650001	21022202650002
17210021000024	35022402252000	35300021002007	33022620000300
36300002702007	33200002260300	35022702252000	35300021002007

0040-

33023020000300	36300002702007	33200002300300	35022402312000
35300021002007	33023220000300	36300002702007	33200002320300
35022702312000	35300021002007	33023320000300	36300002702007
33200002330300	36022702312000	35300021002007	33023020000300
36300002702007	33200002300300	36023402252000	35300021002007
33023520000300	36300002702007	33200002350300	27023602372000
35300021002007	33024020000300	36300002702007	33200002400300
27023602412000	35300021002007	33024220000300	36300002702007

0100-

33200002420300	30023602372000	35300021002007	33024320000300
36300002702007	33200002430300	30023602412000	35300021002007
33024420000300	36300002702007	33200002440300	32024502462100
35300021002007	33024720000300	36300002602007	33200002470300

0120-

31023702512000	35300021002007	34044021000370	31024102512000
35300021002007	34044221000370	26025302532000	35300021002007
34044421000370	26025402532000	35300021002007	34044621000370
26025302542000	35300021002007	34044621002007	26025402542000
35300021002007	34044421000370	25025402542000	35300021002007
34046721000370	26025502552000	35300021002007	34045021000370
25025502552000	35300021002007	34045121000370	24025602572000
35300021002007	34045321000370	24026002572000	35300021002007
34045521000370	24025602612000	35300021002007	34045721000370
24026002612000	35300021002007	34046121000370	23026002612000

35300021002007	34047121000370	24026202572000	35300021002007
34046321000370	23026202572000	35300021002007	34046521000370
0200-			
1720202100f	22fff	f	123456765432
02112233445566	d123456789	-112233445	o01f0001f
10fff	0104432126f	0331704401f	343666324175
607766417164	366232414474	643663324573	604475456334
027777ff	017777ff	310422221111	02401311110211
123456701234	333344445555	457023347011	02123456701234
207665544321	02333344445555	02207665544321	02457023347011
35123456701234	35570112233457	12f1234f	14
0240-			
1234ff	02ff0014	12f1234	121234ff
12ff1234	012302460415	111122224444	010102020404
1	2f0001	1f	111111111111
02111111111111	11111100f	123456701235	247135602470
02123456701235	02247135602470	123456701247	f
f	2fff	10f	3fff
2f	17204021000271	30200703340335	35033502500335
32033503370330	35033504400335	34300021000304	
0300-			
17204021000300	30200703340335	35033503360335	32033503370330
36033503440335	32033503370327	30033503400335	32033503410320
32033503410311	35f21000335	32033503410321	30033503420335
32033503410322	30200703420335	32033503410317	32f03410324
21f02650001	21f02650001	21f02650001	21200002650001
21f02650001	32025003370323	32025003370324	1721002100f
3430002100f			
0334-			
2ff0014	f	3	7777
30	7777ff	14	7777f
5			
0350-			
17204021000350	21000702650001	21022002650001	17210021000007
34300021000010	17204021000355	21001502650001	21022202650002
17210021000015	34300021000016	17204021000362	21002302650001
21022202650002	17210021000023	34300021000022	f
30200703340335	35033502500335	32033503370414	30200703420335
32033503410375	32f03410406	30040603340335	32033503430407
35033502520335	32033503430410	30033503420335	32033503410411
35032302500323	35032402500324	33f20000271	332000f0271
332001f0271	33f20010271	36032302500323	36032402500324
3430002100f			
0440-			
2	4000f2000	02ff0026	4000f2000
543207654321	012345701234	02543207654321	02012345701234
f	012345654321	012345654321	012346570130
4000f0003	02012346570130	024000f0003	012346570130
024000f0003	02012346570130	4000f0003	162752051710
4000f0041	4000f0042	4000f0041	012345701235
012345701234	4000f0003	4000f0003	

Tape Unit Read-Write

Address	Inst.	m ₁	m ₂	m ₃	Remarks
<u>WRITE ADDRESSES:</u>					
0100	21	0123	0122	0003	
0101	34	3000	2100	0127	
0102	15	2100	0000	0000	Write
0103	30	2100	0117	0126	Delay
0104	17	2020	2100	0120	Test for "halt"
0105	17	2100	2100	0102	Test for "repeat order"
0106	35	2000	0116	2000	Modify address
0107	34	2000	0115	0133	Test for end
0110	34	3000	2100	0102	Return
0111	21	2000	2100	0001	Print last address +1
0112	17	2010	2100	0114	Test for halt at end of section
0113	22	f	f	f	
0114	34	3000	2100	0150	
0115	00	0000	0777	0000	Final address
0116	00	0000	0001	0000	
0117	00	0000	0000	0200	
0120	22	f	f	f	
0121	34	3000	2100	0105	
0122	10	0000	0000	0000	
0123	00	3451	6360	7545	
0124	00	6441	4747	6345	
0125	00	4444	4544	6434	
0126	Temp.				
0127	35	2100	2100	2000	
0130	15	2100	0000	0000	Write extra zero
0131	30	2100	0117	0126	Delay
0132	34	3000	2100	0102	
0133	35	0136	2100	2000	
0134	15	2100	0000	0000	Write 17777
0135	34	3000	2100	0111	
0136	00	0001	7777	0000	

If used alone, fill 0115 with highest tape address to be used

READ ADDRESS:

0150	21	0206	0205	0003	
0151	35	2100	2100	2000	Restore address
0152	14	2100	0000	0000	Search
0153	16	2100	0000	2001	Read
0154	30	2100	0176	0126	Delay
0155	17	2100	2100	0170	Test for "repeat order"
0156	33	2001	2000	0165	Check address
0157	33	2000	2001	0165	Check address
0160	17	2020	2100	0177	Test for halt
0161	17	2040	2100	0170	
0162	35	2000	0167	2000	Modify address
0163	34	2000	0115	0201	Test for end
0164	34	3000	2100	0153	
0165	21	2000	2100	0002	Print correct and incorrect address
0166	34	3000	2100	0160	Return
0167	00	0000	0001	0000	
0170	27	2000	0174	0126	
0171	32	0126	0175	0172	
0172	14	2100	000(0	0000)	
0173	34	3000	2100	0153	
0174	02	0000	0000	0014	
0175	00	0000	0001	7777	
0176	00	0000	0000	0200	
0177	22	f	f	f	
0200	34	3000	2100	0161	
0201	21	2000	2100	0001	Print last address
0202	17	2010	2100	0204	Test for halt at end of section
0203	22	f	f	f	
0204	34	3000	2100	0230	
0205	10	0000	0000	0000	
0206	00	3463	4541	4764	
0207	00	4147	4763	4544	

0210 00 4445 4464 6434

If used alone, fill 0115

WRITE MODE 1 OR 2:

0230	34	0310	0274	0272	Test mode
0231	35	0312	2100	0316	Restore to print mode
0232	21	0314	0275	0003	
0233	14	2100	0000	0000	Search
0234	35	2100	2100	0317	Clear address
0235	34	0311	0274	0264	Test pattern
0236	34	3000	2100	0267	Skip
0237	35	(0306)	2100	(2000)	Load Buffer
0240	35	0237	0276	0237	Load Buffer
0241	34	0320	0237	0237	Load Buffer
0242	34	3000	2100	0322	
0243	15	2100	000(1)	0000	Write
0244	30	2100	0277	0126	Delay
0245	17	2020	2100	0252	Test for halt
0246	17	2100	2100	0254	Test for "repeat order"
0247	35	0317	0274	0317	Modify address
0250	34	0317	0115	0260	Test for end
0251	34	3000	2100	0243	
0252	22	f	f	f	Halt
0253	34	3000	2100	0246	Return
0254	27	0317	0300	0126	Shift address
0255	32	0126	0301	0256	Extract address
0256	14	2100	000(0	0000)	Search
0257	34	3000	2100	0243	Return
0260	21	0317	2100	0001	Print last address
0261	17	2010	2100	0263	Test for halt at end of section
0262	22	f	f	f	
0263	34	3000	2100	0350	
0264	35	0302	2100	0237	Restore to load buffer
0265	35	0303	2100	0320	
0266	34	3000	2100	0237	

0267	35	0304	2100	0237	Restore to load buffer
0270	35	0305	2100	0320	
0271	34	3000	2100	0237	
0272	35	0313	2100	0316	Restore to print mode
0273	34	3000	2100	0232	
0274	00	0000	0001	0000	
0275	10	0000	0000	0000	
0276	00	0000	0000	0001	
0277	00	0000	0000	0200	
0300	02	0000	0000	0014	
0301	00	0000	0001	7777	
0302	35	0306	2100	2000	
0303	35	0306	2100	2010	
0304	35	0307	2100	2000	
0305	35	0307	2100	2010	
0306	20	7417	0360	7417	
0307	10	0360	7417	0360	
0310	00	0000	000(1)	0000	Mode
0311	00	0000	000(2)	0000	Pattern
0312	00	7164	6464	6434	
0313	00	0064	6464	6434	
0314	00	3451	6360	7545	
0315	00	6476	7747	4564	
0316	00	(71)64	6464	6434	
0317	Temp.	(address)			
0320	Temp.				
0321	00	0000	0003	0000	
0322	35	0331	2100	0324	
0323	27	0324	0332	0126	
0324	32	0126	0333	200(0)	
0325	35	0324	0276	0324	
0326	34	0334	0324	0323	
0327	32	0310	0321	0243	
0330	34	3000	2100	0243	

0331	32	0126	0333	2000
0332	00	0000	0000	0044
0333	07	0000	0000	0000
0334	32	0126	0333	2010

If used alone, fill 0115, 0310 and 0311

READ MODE 1 or 2:

0350	34	0310	0406	0443	Test mode
0351	35	0441	2100	0437	Restore to print mode
0352	21	0436	0456	0003	
0353	14	2100	0000	0000	Search
0354	35	2100	2100	0317	Clear address
0355	34	0311	0406	0433	Test pattern
0356	35	0431	2100	0460	Restore pattern for check
0357	32	0310	0406	0360	Restore mode
0360	16	2100	000(1)	0000	Read
0361	30	2100	0407	0126	Delay
0362	17	2100	2100	0414	Test for "repeat order"
0363	35	0430	2100	0364	Restore to check
0364	35	(2000)	2100	0461	
0365	34	0460	0461	0422	Check
0366	34	0461	0460	0422	Check
0367	35	0364	0412	0364	
0370	34	0413	0364	0364	
0371	34	3000	2100	0374	Check
0372	33	0461	0462	0422	Check
0373	35	0364	0412	0364	Modify
0374	05	2100	2100	2100	Test for end of check
0375	17	2020	2100	0410	Test for halt
0376	17	2040	2100	0414	
0377	35	0317	0406	0317	Modify address
0400	34	0317	0115	0402	Test for end
0401	34	3000	2100	0360	Return
0402	21	0317	2100	0001	Print last address
0403	17	2010	2100	0405	Test for halt at end of section

0404	22	f	f	f	
0405	34	3000	2100	0445	Go to change pattern, mode
0406	00	0000	0001	0000	
0407	00	0000	0000	0000	
0410	22	f	f	f	
0411	34	3000	2100	0376	
0412	00	0001	0000	0000	
0413	35	2010	2100	0461	(0374)
0414	27	0317	0420	0126	Block search
0415	32	0126	0421	0416	Block search
0416	14	2100	000(0	0000)	Block search
0417	34	3000	2100	0360	Block search
0420	02	0000	0000	0014	
0421	00	0000	0001	7777	
0422	21	0317	2100	0001	Print address
0423	21	0460	2100	0001	Print pattern
0424	21	2000	2100	0010	Incorrect information
0425	21	0435	0456	0001	2 CR's
0426	34	3000	2100	0375	Return
0427	02	0000	0000	0006	
0430	35	2000	2100	0461	(0363)
0431	10	0360	7417	0360	
0432	20	7417	0360	7417	
0433	35	0432	2100	0460	
0434	34	3000	2100	0357	
0435	00	6464	6464	3434	
0436	00	3463	4541	4764	
0437	00	7677	4745	64(71)	
0440	00	6464	6464	6434	
0441	00	7677	4745	6471	
0442	00	7677	4745	6400	
0443	35	0442	2100	0437	
0444	34	3000	2100	0352	
0445	34	0310	0406	0450	Test mode

0446	35	0310	0310	0310	
0447	34	3000	2100	0230	
0450	35	0406	2100	0310	
0451	34	0311	0406	0454	
0452	35	0311	0311	0311	
0453	34	3000	2100	0150	Go to read address
0454	35	0406	2100	0311	
0455	34	3000	2100	0230	
0456	10	0000	0000	0000	
0457	00	0000	0003	0000	
0460	Temps				
0461	Temps				
0462	Temps				

CONSECUTIVE BLOCK SEARCH & ADDRESS CHECK

START 1700: 2010 DOWN, 2020 UP, 2040 DOWN

Address	Inst.	m ₁	m ₂	m ₃	Remarks
1700	35	1743	2100	1704	Fix up for start
1701	35	2100	2100	1745	Fix up for start
1702	35	1744	2100	1713	Fix up for start
1703	35	1746	2100	1747	Fix up for start
1704	14	f	f	f	B.S.
1705	17	2010	2100	1711	Skip the test if up (2010)
1706	16	f	f	f	Read Address
1707	34	2000	1745	1731	Check
1710	34	1745	2000	1731	Check
1711	35	1704	1750	1704	Modify B.S.
1712	35	1745	1751	1745	Modify Check #
1713	14	f	f	0777	B.S.
1714	17	2010	2100	1720	Skip the test if up (2010)
1715	16	f	f	f	Read Address
1716	34	2000	1747	1735	Check
1717	34	1747	2000	1735	Check
1720	36	1713	1750	1713	Modify B.S.

1721	36	1747	1751	1747	Modify Check #
1722	17	2020	2100	1725	Print at next 2 B.S. Com- mands if Down
1723	21	1704	1730	0001	Print B.S. #1
1724	21	1713	1730	0001	Print B.S. #2
1725	34	1713	1704	1704	Test for End
1726	17	2100	2100	1700	Repeat entire test if up
1727	21	1752	1753	0001	Print "End"
1730	22	f	f	f	
1731	21	1704	2100	0001	Error Print #1
1732	21	2000	2100	0001	Error Print #1
1733	17	2040	2100	1704	Repeat error search if up
1734	34	3000	2100	1711	Carry on
1735	21	1713	2100	0001	Error Print #2
1736	21	2000	2100	0001	Error Print #2
1737	17	2040	2100	1713	Repeat error search if up
1740	34	3000	2100	1720	Carry On
1741		-			Slack
1742		-			Slack
1743	14	f	f	f	Const.
1744	14	f	f	0777	Const.
1745		-			Check #1
1746			0777	f	Const.
1747		-			Check #2
1750				1	Const.
1751			1	f	Const.
1752	00	3436	4566	4732	"End"
1753	10	f	f	f	Printer

IBM PROGRAM

Address	Inst.	m ₁	m ₂	m ₃	Remarks
0000	35	2100	2100	0006	
0001	35	f	0004	f	
0002	34	0005	f	f	Clear first 4 channels
0003	11	f	f	f	
0004				1	
0005		2100	2100	0400	
S					
stop code					
0000	21	0200	0140	0002	Print "IBM Test"
0001	21	0202	0141	0001	
0002	35	0040	0011	0011	Take Memory Sum
0003	35	0002	0010	0002	
0004	34	0007	0002	0002	
0005	34	3000	2100	0012	
0006				f	
0007		0400	0011	0011	
0010		1	f	f	
0011				f	Memory sum
0012	27	0377	0020	0022	
0013	35	0022	0021	0021	
0014	36	0012	0010	0012	Take shifted sum
0015	34	0017	0012	0012	
0016	34	3000	2100	0024	
0017		0037	0020	0022	
0020	02	f	f	0017	
0021				f	Shifted sum
0022				f	
0023				f	
0024	33	0011	0033	0035	Test sum
0025	33	0033	0011	0035	

0026	33	0021	0034	0035	Test shifted sum
	33	0034	0021	0035	
0030	21	0203	0140	0003	
	21	0207	0142	0001	Print "Program filled OK."
	34	3000	2100	0040	
0033				f	Correct sum
0034				f	Correct shifted sum
0035	21	0203	0140	0004	
0036	21	0202	0141	0001	Print "Program filled wrong"
0037	22	f	f	f	
0040	05	3000	3000	0104	
0041	13	0174	0171	f	
0042	35	0010	0144	0144	Punch 10 octal cards
0043	30	1777	0143	1777	
0044	13	0174	0171	0004	
0045	35	0010	0144	0144	
0046	34	0146	0144	0041	
0047	35	2100	2100	0144	
0050	05	3000	3000	0114	
0051	12	0175	0173	f	Punch 10 # decimal cards
0052	35	0010	0144	0144	
0053	30	1777	0143	1777	
0054	12	0175	0173	0004	
0055	35	0010	0144	0144	
0056	34	0146	0144	0051	
0057	35	2100	2100	0144	
0060	34	3000	2100	0061	
0061	05	3000	3000	0124	Punch 5 decimal cards
0062	12	0175	0173	0004	
0063	35	0010	0144	0144	
0064	35	0010	0144	0144	
0065	34	0146	0144	0062	

0066	35	2100	2100	0144	
0067	34	3000	2100	0071	
0070				f	
0071	17	2010	3000	0040	Punch another set if 2010 up
0072	22	f	f	f	
PART II					
0073	05	3000	3000	2100	Jump to read if restarted
0074	34	3000	2100	0210	
0100			1	f	
0104				f	
0105				f	
0106	22	2222	2222	2222	
0107				f	
0110	04	4444	4444	4444	
0111				f	
0112	26	6666	6666	6666	
0113				f	
0114	+		f	f	
0115				f	
0116	-	222	222	222	
0117				f	
0120	+	444	444	444	
0121				f	
0122	-	666	666	666	
0123				f	
0124	+	888	888	888	
0125				f	
0126	-	888	888	888	
0140	10	f	f	f	
0141	10	0001	f	f	
0142	10	f	0001	f	
0143				200	
0144				f	
0145	11	1111	1111	1111	

0146		12	f	f	
0147	33	3333	3333	3333	
0150		5	f	f	
0151	15	5555	5555	5555	
0152				f	
0153	37	7777	7777	7777	
0154				f	
0155	+	111	111	111	
0156				f	
0157	-	333	333	333	
0160				f	
0161	+	555	555	555	
0162				f	
0163	-	777	777	777	
0164				f	
0165	+	999	999	999	
0166				f	
0167	-	999	999	999	
0170	15	5555	5555	5556	
0171	11	1111	1111	1111	
0172	01	6735	6735	6736	
0173	01	0421	0421	0421	
0174	26	6666	6666	6667	
0175	02	7356	7356	7357	
0176				f	
0177	02	f	f	0014	
0200		3436	6057	7664	
0201		7545	4475	4432	
0202		3234	f	f	
0203		3436	7063	7773	Alphabetic Codes
0204		6341	7664	4660	
0205		7171	4547	6464	
0206		5163	7766	7364	
0207		7742	3234	f	
0210	06	0170	0171	f	

0211	30	1777	0143	1777	
0212	06	0170	0171	0004	Read and Check
0213	04	3000	3000	0310	10 octal cards
0214	05	3000	3000	0104	
0215	04	3000	3000	0300	
0216	35	3000	2100	0320	
0217	34	3000	2100	0247	
0220	35	0010	0144	0144	
0221	34	0150	0144	0210	
0222	35	2100	2100	0144	
0223	06	0172	0173	f	
0224	30	1777	0143	1777	Read and Check
0225	06	0172	0173	0004	10 decimal cards
0226	04	3000	3000	0310	
0227	05	3000	3000	0114	
0230	04	3000	3000	0300	
0231	35	3000	2100	0320	
0232	34	3000	2100	0247	
0233	35	0010	0144	0144	
0234	34	0150	0144	0223	
0235	35	2100	2100	0144	
0236	05	3000	3000	0124	
0237	06	0172	0173	0004	
0240	04	3000	3000	0310	
0241	05	3000	3000	0124	Read 5 decimal cards
0242	04	3000	3000	0300	
0243	35	3000	2100	0320	
0244	34	3000	2100	0247	
0245	34	3000	2100	0360	
0246				f	
0247	05	3000	3000	2100	
0250	35	0320	0100	0320	
0251	32	0320	0176	0105	
0252	30	0105	0177	0105	

0253	32	0105	0101	0274	
0254	33	0300	0310	0275	
0255	33	0310	0300	0275	
0256	33	0341	0351	0275	Check Subroutine
0257	33	0351	0341	0275	
0260	33	0302	0312	0275	
0261	33	0312	0302	0275	
0262	33	0343	0353	0275	
0263	33	0353	0343	0275	
0264	33	0304	0314	0275	
0265	33	0314	0304	0275	
0266	33	0345	0355	0275	
0267	33	0355	0345	0275	
0270	33	0306	0316	0275	
0271	33	0316	0306	0275	
0272	33	0347	0357	0275	
0273	33	0357	0347	0275	
0274	34	3000	2100	f	Return ink
0275	05	3000	3000	0310	
0276	34	3000	2100	0321	
0277	22	f	f	f	

Fill Test Pattern

This program fills test pattern into all main memory cells.

```
-o2000 2000
    35 2004 2100 0000
    35 2000 2005 2000
    34 2003 2000 2000
    22 2004 2100 2000
    25 2525 2525 2525 (or any test pattern)
    1
```

Compute. Computer will halt when main memory is filled with test pattern. Time: 37 sec.

Memory sum = 25252525252000

Erase Main Memory

This program uses a double putaway command to insure that $P_2 O_{13}$ pulses are erased.

```
-o2000 2000-
    26 2100 2100 0000
    35 2000 2004 2000
    34 2003 2000 2000
    22 2100 2100 2000
    1
```

Compute. Computer will halt when main memory is erased (cells 0000 to 1777). Time: 30 sec.

Sum and Shifted Sum of Main Memory = 0

Compare all Channels with Channel 0**NOTE**

Before running this program, the Fill Test Pattern Program should be filled into the memory.

```
-o0061 0061
    37 0026 0142 0062
    17 0027 0076 0061
    17 2100 2100 0063
    35 0061 0066 0061
```

```
34 0067 0061 0061
```

```
0100 f
```

```
0026 2000 f
```

Throw Test switch 2020 up, 2100 up, and depress COMPUTE button.

Throw 2020 down-up, then 2100 down-up, this advances the comparison one channel. Error print occurs after Channel 17 comparison. To compare each channel with Channel 0, observe channel switching in Program Counter Block 14.

Minimum Access Test

```
-o0045-
```

```
30 2100 2100 0703
```

```
35 0711 0054 1603
```

```
36 1603 1601 1602
```

```
34 1602 2100 0053
```

```
17 2010 3000 0045
```

```
22 2100 2100 2100
```

```
21 1603 0055 0001
```

```
11 2233 4567 6543
```

```
02 f f f
```

```
0711-11 2233 4567 6543
```

```
00451601-11 4467 1357 5306
```

Throw switch 2010 up. Compute (Sync scope on PC 134 to observe memory waveform.)

Memory Sum

This program sums memory cells 0000 to 1777 and prints out the sum.

```
20002000-
```

```
2000 35 0000 2007 2007
```

```
2001 35 2000 2006 2000
```

```
2002 34 2005 2000 2000
```

```
2003 21 2007 2100 0001
```

```
2004 22 f f f
```

```
2005 2000 f f
```

```
2006 1 f f
```

```
2007 o
```

Throw Auto Overflow Test switch to OUT Compute. Printout = Memory Sum. Time 40 sec.

Shifted Memory Sum

This program sums memory cells 0000 to 1777 and prints out sum. Before summing, each cell is shifted so that sign digits are in center of word. This sum gives a check on correctness of sign digits; whereas, the sum obtained with the Memory Sum program does not provide such a check.

20002000-

2000	27	1776	2005	1777
2001	35	1777	2007	2007 (tab)
2002	36	2000	2006	2000
2003	34	2001	2000	2000
2004	21	2007	2100	0001
2005	22	0000	0000	0017
2006	1	f	f	
2007			0	

Compute. Printout = Shifted Memory Sum. Time: 100 sec.

NOTE

This program uses cell 1777, but the sum of the contents of the remaining cells is not affected by this use. If 1777 contains part of program being summed, it will not be included in the sum, and must be refilled.

Component Installation and Location Charts

PLUG-IN UNIT CROSS-REFERENCE LIST

<u>Assembly No.</u>	<u>Schematic No.</u>	<u>Function</u>	<u>Tube(s)</u>
00267001	00263002	Logical Driver	5687
00367001	00363002	Flip-Flop Driver	5687
00467001	00463002	Logical Driver	5963
00567001	00563002	Flip-Flop Driver	5963
00667001	00663002	Flip-Flop Driver	5881
10567002	10563001	One-Shot, Unclamped	12AT7
10667016	10663006	Dual Phase Inverter	5687
10667017	10663007	Clipper	6BQ7-A
10667018	10663008	Schmitt Trigger	12AT7
10667029	10663028	One-Shot, Clamped	5687
12667102	12663101	Relay Driver	2-6BQ7-A
12767157	12763156	Delay Line and Cathode Follower]	5963
20267002	20263001	Flip-Flop, Clamped	12BH7
20267004	20263003	Recirculating Line Record Circuit]	5687
20267006	20263005	Preamplifier	6BQ7-A
20267010	20263009	Cathode Follower	2-12AT7
20267012	20263011	Flip-Flop, Clamped	12AT7
20267014	20263013	Quintuple Input Coupler	None
20267179	20263178	Flip-Flop, Unclamped	12AT7
20267293	20263292	Cathode Follower Driver	5881
20267339	20263329	Main Memory Record- Playback Circuit]	2-6BQ7-A 2-5963
20267340	20263337	Main Memory Limiter Amplifier]	1-12AT7 1-5687
20267365	20263364	Special Record Tube Driver	2-6AN5
20267367	20263366	Special Preamplifier	6BQ7A
20267396	20263395	Special Cathode Follower]	2-12AT7

PLUG-IN UNIT LOCATIONS BY CHASIS

(Assembly Nos. Shown - For Schematic Nos. See Plug-in Unit Cross-Reference)

TUBE CHASSIS NO. 1 (TC-1)

	D	C	B	A
1	10667017 F Clipper	10667016 F, G Ph. Inv.	Blank	00267001 J ₁ , J ₁ ' Dr.
2	20267012 F ₁ , F ₁ ' F.F.	00367001 F ₂ , F ₂ ' Dr.	20267179 F ₂ Flip-Flop	10667017 G Clipper
3	00467001 H ₂ K'+H'K (in G ₀)	00467001 F' ₀ G' ₀ Dr.	00467001 G ₁ , G ₁ ' Dr.	20267012 G ₁ Flip-Flop
4	20267179 G ₅ Flip-Flop	20267012 G ₄ Flip-Flop	20267012 G ₃ , G ₃ ' F.F.	20267012 G ₂ , G ₂ ' F.F.
5	00367001 G ₅ , G ₅ ' Dr.	00467001 G ₄ , G ₄ ' Dr.	00467001 F ₂ A' ₂ +F' ₂ A ₂	20267010 G ₀ , F ₀
6	10667017 H Clipper	10667016 H, J Ph. Inv.	12667102 Tape Unit Mode	20267010 G ₀ , H ₀
7	20267012 H ₁ , H ₁ ' F.F.	20267179 H ₂ Flip-Flop	00367001 H ₂ , H ₂ ' Dr.	00467001 H' ₀ , J' ₀ Dr.
8	00367001 J ₂ , J' ₂ Dr.	20267179 J ₂ Flip-Flop	20267012 J ₁ Flip-Flop	10667017 J Clipper
9	10667018 Clock Schmitt	12767157 Clock Delay	00267001 C _s	00667001 C
10	20267293 Sum Snapper	00467001 SS Driver	20267010 S, J ₀	20267010 J ₀
11	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)
12	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)
13	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)
14	20267365 False Record	20267365 True Record	00467001 J ₀ (IBM)	00467001 J ₀ (IBM)
15	Blank	20267012 Start F.F.	10667029 SS O.S. Cl.	10567002 Start O.S. Uncl.

TUBE CHASSIS NO. 2 (TC-2)

	A	B	C	D
1	20267179 A ₁ Flip-Flop	00667001 A ₁ Driver	00667001 A ₁ ' Driver	00267001 A ₁ A ₂ +A ₁ 'A ₂ '
2	00267001 E ₅ F ₂ +E ₅ 'F ₂ '	20267010 a ₁ , E ₀	00267001 O ₁₂₋₁₃ A ₄ A ₆ '+A ₄ 'A ₆	00267001 O ₄₋₇ , O ₀₋₃
3	00467001 O ₃ +P ₂ O ₂	00467001 O ₀ , (P ₂ O ₁₁)'	00267001 O ₀ , P ₂ O ₁₁	00267001 P ₀ O ₁₂ , P ₁ O ₁₂
4	20267179 A ₂ Flip-Flop	00667001 A ₂ Driver	00667001 A ₂ ' Driver	20267010 A ₁₀ , A ₁₀ ', A ₉ , A ₁₁
5	20267010 O ₁ ^a , E ₀	20267010 a ₂ , E ₀	20267010 O ₂ ^a , E ₀	00467001 (P ₀ O ₁₂)', (P ₁ O ₁₂)'
6	20267179 A ₃ Flip-Flop	00667001 A ₃ Driver	00667001 A ₃ ' Driver	00467001 O ₀ +P ₀ O ₁
7	20267010 a ₃ , O ₃ ^a	00467001 O ₄ +P ₀ O ₅	00467001 O ₇ +P ₂ O ₆	00467001 O ₈ +P ₀ O ₉
8	20267179 A ₄ Flip-Flop	00667001 A ₄ Driver	00667001 A ₄ ' Driver	12667102 Block Address
9	00467001 O ₁₁ +P ₂ O ₁₀	00467001 (O ₁₂ +P ₀ O ₁₃)(P ₀ O ₀)'	20267010 a ₄ , O ₄ ^a	12667102 Block Address
10	00467001 (P ₁ O ₁₃)', O ₁₃ '	00267001 O ₁₂ +P ₀ O ₁₃ , P ₀ O ₀	20267010 a ₅ , O ₅ ^a	12667102 Block Address
11	20267179 A ₅ Flip-Flop	00367001 A ₅ , A ₅ ' Dr.	00467001 O ₅ ^a , O ₆ ^a	00267001 P ₁ O ₁₃ , O ₁₃
12	20267179 A ₆ Flip-Flop	00367001 A ₆ , A ₆ ' Dr.	00267001 P ₂ O ₁₃ , P ₂ O ₁₃ '	00267001 O ₀₋₁₁
13	20267179 A ₇ Flip-Flop	00367001 A ₇ , A ₇ ' Dr.	20267010 a ₆ , O ₆ ^a , a ₇ , O ₇ ^a	00467001 A ₁₂ ' Driver
14	20267012 A ₈ , A ₈ ' F.F.	20267012 A ₁₀ Flip-Flop	20267010 a ₈ , O ₈ ^a	00467001 A ₆ ', A ₅ Dr.
15	20267012 A ₉ , A ₉ ' F.F.	20267012 A ₁₁ Flip-Flop	20267012 A ₁₂ Flip-Flop	20267012 A ₁₃ Flip-Flop

TUBE CHASSIS NO. 3 (TC-3)

	A	B	C	D
1	20267179 R ₁ Flip-Flop	00367001 R ₁ , R ₂ Dr.	00267001 K'(0'+Y')	00467001 X'Driver
2	20267179 R ₂ Flip-Flop	20267179 M Flip-Flop	00567001 M, M' Dr.	00667001 N ₁ Driver
3	00467001 M (Early)	10667016 M Ph. Inv.	10667017 M Clipper	00667001 N ₁ Driver
4	20267012 M (Early)	00667001 N ₁ ' Driver	00667001 N ₂ Driver	00667001 N ₂ Driver
5	20267010 PCSn ₄ , PCSn ₁ PCS29 _{0k}	00667001 N ₂ ' Driver	00667001 N ₃ Driver	00667001 N ₃ Driver
6	20267179 N ₁ Flip-Flop	00267001 X Driver	00667001 N ₃ ' Driver	00667001 N ₃ ' Driver
7	20267179 N ₂ Flip-Flop	00667001 N ₄ Driver	00667001 N ₄ Driver	00667001 N ₄ ' Driver
8	20267179 N ₃ Flip-Flop	00667001 N ₄ ' Driver	00667001 N ₅ Driver	00667001 N ₅ Driver
9	20267179 N ₄ Flip-Flop	20267010 PCSn _{0n1} , PCSn ₃ PCSn _{0n2} , PCSn _{0n4}	00667001 N ₅ ' Driver	00667001 N ₅ ' Driver
10	20267179 N ₅ Flip-Flop	00667001 N ₆ Driver	00667001 N ₆ Driver	00667001 N ₆ ' Driver
11	20267179 N ₆ Flip-Flop	00667001 N ₆ ' Driver	00667001 N ₇ Driver	00667001 N ₇ Driver
12	20267179 N ₇ Flip-Flop	00667001 N ₈ Driver	00667001 N ₇ ' Driver	00667001 N ₇ ' Driver
13	20267179 N ₈ Flip-Flop	20267010 PCSn ₇ , PCSn ₃ , PCSn ₅	00667001 N ₈ Driver	00667001 N ₈ ' Driver
14	20267179 N ₉ Flip-Flop	00467001 PCS5 Dr.	00667001 N ₈ ' Driver	00667001 N ₉ ' Driver
15	20267010 PCSn ₆ , PCSn _{0n6}	00667001 N ₉ Driver	00667001 N ₉ Driver	00667001 N ₉ ' Driver

TUBE CHASSIS NO. 4 (TC-4)

	D	C	B	A
1	00467001 E ₁ , E ₁ ¹ Dr.	20267012 E ₁ Flip-Flop	10667017 E Clipper	01667016 E Ph. Inv.
2	20267010 See PCS Dwg.	20267179 E ₄ Flip-Flop	20267012 E ₃ , E ₃ ¹ F.F.	20267012 E ₂ , E ₂ ¹ F.F.
3	00467001 E ₀ ¹ Driver	00667001 E ₅ ¹ Driver	00667001 E ₅ Driver	20267179 E ₅ Flip-Flop
4	10667029 Punch O.S.	00267001 462+512+532+561	20267010 PCS41 _k , PCS30 _{ok} PCS103 _k , PCS29 _g	20267010 See PCS Dwg.
5	10667029 Reader O.S.	Blank	10667017 M _w Clipper	10667016 M _w Ph. Inv.
6	00567001 M _w , M _w ¹ Dr.	20267179 M _w Flip-Flop	00467001 M _w (Early)	20267010 PCS30 _k , P+Props.
7	Blank	20267012 M _w (Early)	Blank	20267010 PCS _{n5}
8	00367001 D ₂ Driver	00367001 D ₁ Driver	20267010 PCS1, PCS2, PCS3, PCS29 _k	00367001 E ₄ , E ₄ ¹ Dr.
9	00367001 D ₃ Driver	00367001 P ₂ Driver	20267179 K	20267010 PCS _{0n9} , ok
10	00367001 D ₄ Driver	00367001 P ₁ Driver	00367001 K	20267010 k
11	20267179 D ₄ Flip-Flop	20267179 D ₃ Flip-Flop	20267179 P ₂ Flip-Flop	20267179 P ₁ Flip-Flop
12	20267179 D ₂ Flip-Flop	20267014 l ₃ , 0 ¹ ₃ , l ₄ , 0 ¹ ₄	20267179 D ₁ Flip-Flop	20267014 l ₁ , 0 ¹ ₁ , l ₂ , 0 ¹ ₂
13	00567001 L ₃ , L ₃ ¹ Dr.	20267179 L ₃ Flip-Flop	00567001 L ₁ , L ₁ ¹ Dr.	20267179 L ₁ Flip-Flop
14	00267001 C _t	20267179 L ₄ Flip-Flop	20267179 L ₅ Flip-Flop	20267179 L ₂ Flip-Flop
15	20267010 PCS _{n8} , LF.F. Inputs	00567001 L ₄ , L ₄ ¹ Dr.	00567001 L ₅ , L ₅ ¹ Dr.	00567001 L ₂ , L ₂ ¹ Dr.

MEMORY TUBE CHASSIS (MTC)

	<u>MTC 1/2</u>	<u>MTC 3/4</u>	<u>MTC 1/3</u>	<u>MTC 2/4</u>
A	20267006 F Preamp	20267006 G Preamp A	20267006 H Preamp	20267006 E Preamp
B	20267006 J Preamp	20267006 B Clock Preamp	Blank	20267367 M _w Preamp
C	20267339 Ch. 5, Ch. 6	20267339 C Ch. 13, Ch. 14	20267339 Ch. 1, Ch. 2	20267339 Ch. 10, Ch. 17
D	20267339 Ch. 7, Ch. 0	20267339 D Ch. 15, Ch. 16	20267339 Ch. 3, Ch. 4	20267339 Ch. 11, Ch. 12
E	20267339 M _w	20267340 E-1 M Lim. Amp.	20267002 H Rec.F.F.	20267002 J Rec.F.F.
F-1	20267002 E Rec.F.F.	20267002 E-2 G Rec.F.F.	20267004 F, H Record	20267004 G Record
F-2	20267004 E, J Record	20267002 F Rec.F.F.		

UPPER TUBE CHASSIS #2(UTC-2)

	A	B
0	00467001 T ₁ , T ₂ Dr.	00467001 T ₃ , T ₄ Dr.
1	00467001 T ₁ ⁱ , T ₂ ⁱ Dr.	20267396 L ₁ , L ₂ , L ₃ , L ₄
2	12667102 Ry. 25, Ry. 26, Ry. 27, Ry. 30	00467001 T ₃ ⁱ , T ₄ ⁱ Dr.
3	12667102 Tape Unit No.	12667102 Ry. 28, Ry. 31, Ry. 32, Ry. 33
4	12667102 Ry. 37, Ry. 38, Ry. 39, Ry. 40	12667102 Ry. 41, Ry. 42 Ry. 27, Ry. 30

PLUG-IN UNIT COMPLEMENT AND LOCATIONS

<u>Assembly No. *</u>	<u>Qty.</u>	<u>Locations</u>
00267001	16	TC1: A1, B9 TC2: A2, B10, C2, C3, C12, D1, D2, D3, D11, D12 TC3: B6, C1 TC4: C4, D14
00367001	16	TC1: B7, C2, D5, D8 TC2: B11, B12, B13 TC3: B1 TC4: A8, B10, C8, C9, C10, D8, D9, D10
00467001	44	TC1: A7, A11, A12, A13, A14, B3, B5, B11, B12, B13, B14, C3, C5, C10, C11, C12, C13, D3, D11, D12, D13 TC2: A3, A9, A10, B3, B7, B9, C7, C11, D5, D6, D7, D13, D14 TC3: A3, B14, D1 TC4: B6, D1, D3 UTC2: A0, A1, B0, B2
00567001	7	TC3: C2 TC4: A15, B13, B15, C15, D6, D13
00667001	45	TC1: A9 TC2: B1, B4, B6, B8, C1, C4, C6, C8 TC3: B4, B5, B7, B8, B10, B11, B12, B15, C4 through C15, D2 through D15 TC4: B3, C3
10567002	1	TC1: A15
10667016	5	TC1: C1, C6 TC3: B3 TC4: A1, A5
10667017	7	TC1: A2, A8, D1, D6 TC3: C3 TC4: B1, B5
10667018	1	TC1: D9
10667029	3	TC1: B15 TC4: D4, D5
12667102	9	TC1: B6 TC2: D8, D9, D10 UTC2: A2, A3, A4 B3, B4
12767157	1	TC1: C9

*For unit names and schematic nos., see Plug-in Unit Cross-Reference

<u>Assembly No. *</u>	<u>Qty.</u>	<u>Locations</u>
20267002	5	MTC 1/2: F1 MTC 2/3: E1 MTC 2/4: E1 MTC 3/4: F1, F2
20267004	3	MTC 1/2: F2 MTC 2/3: E2 MTC 2/4: E2
20267006	6	MTC 1/2: A, B MTC 2/3: A MTC 2/4: A MTC 3/4: A, B
20267010	27	TC1: A5, A6, A10, B10 TC2: A5, A7, B2, B5, C5, C9, C10, C13, C14, D4 TC3: A5, A15, B9, B13 TC4: A4, A6, A7, A9, A10, B4, B8, D2, D15
20267012	19	TC1: A3, A4, B4, B8, C4, C15, D2, D7 TC2: A14, A15, B14, B15, C15, D15 TC3: A4 TC4: A2, B2, C1, C7
20267014	2	TC4: A12, C12
20267179	38	TC1: B2, C7, C8, D4 TC2: A1, A4, A6, A8, A11, A12, A13 TC3: A1, A2, A6 through A14, B2 TC4: A3, A11, A13, A14, B9, B11, B12, B14, C2, C6, C11, C13, C14, D11, D12
20267293	1	TC1: D10
20267339	9	MTC 1/2: C, D, E MTC 2/3: C, D MTC 2/4: C, D MTC 3/4: C, D
20267340	1	MTC 3/4: E
20267365	2	TC1: C14, D14
20267367	1	MTC 2/4: B
20267396	<u>1</u>	UTC2: B1
	270	Total

*For unit names and schematic nos., see Plug-in Unit Cross-Reference

TUBE COMPLEMENT AND LOCATIONS

<u>Type</u>	<u>In Plug- in Units</u>	<u>Not In</u>	<u>Plug-in</u>	<u>Units</u>	<u>Stabiline LPS-2</u>	<u>Totals</u>
		<u>UTC-1</u>	<u>UPS-3</u>	<u>UPS-4</u>		
OA2				1		1
OB2			1	1		2
2D21		57				57
5U4-G					2	2
6AN5	4					4
6AU6			1	2		3
6BN6					1	1
6BQ7-A	50					50
6L6-G					2	2
6SJ7					1	1
12AT7	116					116
12BH7	5					5
VR-150					1	1
1236					1	1
5687	44					44
5881	46					46
5963	70		1			71
6080			1	3		4
Totals	335	57	4	7	8	411

RELAY CONVERSION CHART

Used in Model	NCR Component Standard No.	Relay, Comar Elec.Co. Part No.	Relay, Globe Elec.Mfg.Co. Part No.	Obsolete Advance Relay Part No.	Relay Description
126	CS-648-2762	C-4650	3C3C153	TC9A110-D3	Plug-in, small pins, 3PDT-AC
126	CS-648-2763	C-4651	4C3C207	TC12A110-D3	Plug-in, small pins, 4PDT-DC
102 & 126	CS-648-2764	C-4652	2C3C208	TM3D60-D1 TM6D60-D1	Plug-in, large pins, 2PDT-DC
102	CS-648-2765	C-4653	2C3C157	-	Solder lugs, 2PDT-DC
126	CS-648-2766	C-4654	2C3C158	TT3D60-D9 TT6D60-D9	Plug-in, large pins, 2PDT-DC
102 & 126	CS-648-2767	C-4655	2C3C159	TT6D110-D9	Plug-in, large pins, 2PDT-DC
102	CS-648-2768	C-4656	4C3C209	TT12D60-A3	Solder lugs, 4PDT-DC
102 & 126	CS-648-2769	C-4657	4C3C210	TT12D110-D3	Plug-in, small pins, 4PDT-DC

Note: Only Comar or Globe relays to be used in 102-A and 126, except use Advance TM6D60-D1 or NCR Component Standard No. CS-648-4193 for relays 29 and 43 through 48 in 102-A only.

FUSE COMPLEMENT

<u>Function</u>	<u>Location</u>	<u>Qty.</u>	<u>Size</u>
Main Line	Behind Fuse Panel	2	50A Cartridge Renewable
Fan and Shutters	Fuse Panel	2	15A Slo-Blo
Drum Motor	Fuse Panel	2	15A 3AG
+225V DC Supply (Primary)	Fuse Panel	2	15A 3AG
+250V DC Supply (Primary)	Fuse Panel	2	2A 3AG
-300V DC Supply (Primary)	Fuse Panel	2	2A 3AG
+90V Relay Supply (Primary)	Fuse Panel	2	2A 3AG
6.3V AC 3 Amp to UPS-3. (Primary)	Fuse Panel	2	1/4A 3AG
6.3V AC 3 Amp to UPS-4. (Primary)	Fuse Panel	2	1/4A 3AG
12.6V AC 100 Amp (Primary)	Fuse Panel	2	8A 3AG
12.6V AC 25 Amp (Primary)	Fuse Panel	2	3A 3AG
+225V DC Load	Fuse Panel	1	8A 3AG
+100V DC Load	Fuse Panel	1	5A 3AG
-300V DC Load	Fuse Panel	1	1/2A 3AG
+100V Regulator	UPS-4	1	1/2A Slo-Blo
+250V DC Load	UPS-3	1	1A 3AG
Motor	Flexowriter	1	3A Slo-Blo
DC	Flexowriter	1	1A 3AG
Line	Stabiline	2	20A Cartridge (Buss Non-20)
AC Regulator (Primary)	Stabiline	1	4A 3AG
Total		<u>32</u>	

SWITCHES AND ADJUSTMENTS

<u>Name</u>	<u>Type</u>	<u>Location</u>	<u>Function</u>
Power On	PB NO*	Console - Control Panel	Starts Computer Power Supply "On" Sequence
Power Off	PB NC	"	If memory safe switch down, turns off computer
Base 10	PB NO	"	Turns on A ₆ Flip-Flop
Base 8	PB NO	"	Turns Off A ₆ Flip-Flop
Compute	PBNO	"	Turns On A ₁ -A ₅ , Generates Start Signal When Released
Clear	PB NO	"	Returns Program Counter to 0 (Rest. Resets A ₁ -A ₆
Memory Safe	T	"	Same as Clear Button (When Down), but also permits "Power Off" Function
2010, 2020, 2040, 2100	T (4)	"	Test Switches Used in "ts" Command
Automatic Overflow Test	T	"	"In" - Overflow Causes Error Print "Out" No Overflow Error Print
Fill	T	"	Up - Permits Filling From Flexowriter
On-Off	T	Flexowriter	Turns On Flexowriter Motor and Power Supply
DC On	PB NO	Test Panel	Applies DC to Computer Circuits
DC Off	PB NO	"	If Test Panel Memory Safe Switch Down, Removes DC from Circuits
Memory Safe	T	"	Same as Clear Button, but also permits "DC Off" Function
YN ₁ -YN ₉	T (9)	"	9-bit switch register for Y test No. UP = 1

*PB = Push Button; NO = Normally Open; NC = Normally Closed;
T = Toggle; R = Rotary Tap; TM = Toggle-Momentary

SWITCHES (Cont'd)

Y	T	Test Panel	UP - Permits Y Test Function
YK-K'	T	"	Sets K in Y Test Function
XN ₁ -XN ₉	T (9)	"	9-bit switch register for X test No. UP = 1
X	T	"	UP - Permits X Test Function
Meter Selector	R	"	Connects Meter and Appropriate Multiplier to Voltage Selected
S1**	TM	UPS-3	Permits +225V Load Current Measurement at J1 and J2
S2**	TM	UPS-3	Permits +250V Load Current Measurement at J3 and J4
S3**	TM	UPS-4	Permits +100V Regulator Current Measurement at J7 and J8
S4**	TM	UPS-4	Permits -300V Load Current Measurement at J9 and J10
S5**	TM	UPS-4	Permits +100V Load Current Measurement at J5 and J6
T8 Tap Switch	R	LPS-4	Adjusts +90 Sequence Relay Voltage
S6	T	Below TS4-3	Turns Off Drum Motor
S7	T	Below TS4-401	Turns Off DC - Prevents "D C On" Button Function
SWSP-WD	T	Below TS3-3	Permits Word Channel Recording, Disables Channel Selector Matrix
Stabiline	T	LPS-2	Turns Off Stabiline. WARNING: Never Throw Switch Down with Computer On.

** CAUTION: Never throw these switches unless current meter is connected in circuit.

ADJUSTMENTS

<u>Name</u>	<u>Location</u>	<u>Correct Adjustment</u>
Fan Thermostat	Fan Contactor Bracket	1 Turn CCW from Ambient Temp. Closing Point
Overheat Thermostat	"	1 Turn CCW from Ambient Temp. Opening Point
R58 and R107	"	Set for Minimum Change In +100V from DC Off to DC On Condition. Keep both taps same distance from End. Approximately 66 ohms each.
R64 and R108	"	See "Power Supplies", Section IV.
R46, 500 50 W Adjustable (Shunts-300v regulator tube)	UPS-4	With Computer On, Turn DC Off, Remove Wire from Slider and Insert 0-500 Milliammeter, (Negative Lead to Slider). Turn on DC and Note Current. Adjust (Turning Off DC to Move Slider) until Meter Reads 240-260 ma. Reconnect Wire to Slider, (Required only if Resistor Replaced.)
R51 (-300)	UPS-4	See "Power Supplies", Section IV
R40 (+100)	UPS-4	See "Power Supplies", Section IV
R19 (+250)	UPS-3	Adjust for Minimum Ripple at J3 (Do <u>not</u> throw S2)
R21 (NCB)	UPS-3	Set Meter Switch to "NCB". Adjust until Needle is in +5% Region
R30(Memory Clipper Bias)	UPS-3	Adjust for +24V between TS 3-590 and Ground
R31 (GCB)	UPS-3	Set Meter Switch to "GCB" or "LGC". Adjust until Needle is in <u>±</u> 5% Region.
Stabiline Overvoltage Adj.	Under Knurled Cap on Back of LPS-2	Disconnect Load from Stabiline. Attach AC Voltmeter to Output Terminals. Turn Stabiline Output Adj. to Full CW Position. (Maximum Output). Turn Overvoltage

Adjustment CCW until Output Voltage just starts to drop. Turn Stabiline Output Adj. to Minimum (CCW), Reconnect Load Adjust +225 per instructions under "Power Supplies", Section IV.

Stabiline
Output Adj.

Under Knurled
Cap - LPS-2

See "Power Supplies", Section IV.

For Gain and DC Level Adjustments in Main Memory, Recirculating Registers, Clock, and Word Channels, See Section III.

Spare Parts

Plug-In Units

<u>Quantity</u>	<u>Assembly No.</u>
1	00167001
1	00267001
1	00367001
1	00467001
1	00567001
1	00667001
1	10567002
1	10667017
1	10667029
1	12667106
1	12667146
1	12667289
1	12667328
1	12667332
1	20267002
1	20267012
1	20267179
1	20267365
1	20267367
1	12667368
1	20267404

Tubes

1	OA2
1	OB2
4	2D21
1	5R4
2	5U4G
2	6AB4
2	6AN5
1	6AS5
2	6AU6
2	6BN6
20	6BQ7A
1	6SJ7
2	6U8
30	12AT7

Tubes (Cont'd)

<u>Quantity</u>	<u>Assembly No.</u>
4	12AX7
3	12AY7
2	12BH7
1	1236A (Superior Electric)
1	5651
10	5687
4	5881
6	5963
4	6080
2	5965

<u>Quantity</u>	<u>Description</u>	<u>Part Number</u>	<u>Manufacturer</u>
Capacitors			
5	22 mmfd 500v mica	GM-15	El Menco
5	50 mmfd 500v mica	GM-15	El Menco
5	100 mmfd 500v mica	GM-15	El Menco
	.001 mfd 600v hermetically sealed - tubular	81P10206-S1	Sprague
5	.01 mfd 600v hermetically sealed - tubular	81P10306-S3	Sprague
5	.1 mfd 400v hermetically sealed - tubular	78P10404-S3	Sprague
1	1mfd 400v molded plastic tubular - Tiny Chief	PJ4W1	Sprague
1	300 mfd 450v electrolytic	HC-45003	Mallory
Diodes			
100	F (G.E. Type A)		Trans.
50	G ₁ (G.E. Type D)		Trans.
25	Type E		General Electric
Switches			
1	Snap action	BBp-2	Tyni Switch
1	Contact, signal	888-4409	
1	Rotor, signal	888-4410	
Magnetic Head and Mount Assemblies			
1		14137001	NCR
1		14137006	NCR
1		14237009	NCR

SPARE PARTS

G-3

<u>Quantity</u>	<u>Description</u>	<u>Part Number</u>	<u>Manufacturer</u>
Pilot Lights			
4	6-8v	44	General Electric
10	Neon	NE-51	General Electric
Relays (see Relay Conversion Chart, Page F-10)			
2		C-4650	Comar
2		C-4651	Comar
2		C-4652	Comar
2		C-4653	Comar
2		C-4654	Comar
4		C-4655	Comar
2		C-4656	Comar
2		C-4657	Comar
2		TM3D60-D1	Advance
4		TM6D60-D1	Advance
Resistors			
6	40K 1W 1% Precision		Welwyn
6	100K 1W 1% Precision		Welwyn
6	120K 1W 1% Precision		Welwyn
6	130K 1W 1% Precision		Welwyn
6	150K 1W 1% Precision		Welwyn
6	200K 1W 1% Precision		Welwyn
6	250K 1W 1% Precision		Welwyn
6	280K 1W 1% Precision		Welwyn
6	300K 1W 1% Precision		Welwyn
6	450K 1W 1% Precision		Welwyn
6	500K 1W 1% Precision		Allen Bradley
5	100 ohm 1/2W 5%		Allen Bradley
5	330 ohm 1/2W 5%		Allen Bradley
5	1K 1/2W 5%		Allen Bradley
5	3.3K 1/2W 5%		Allen Bradley
5	10K 1/2W 5%		Allen Bradley
5	33K 1/2W 5%		Allen Bradley
5	100K 1/2W 5%		Allen Bradley
5	330K 1/2W 5%		Allen Bradley
5	1 meg 1/2W 5%		Allen Bradley
5	10K 1W 5%		Allen Bradley
5	33K 1W 5%		Allen Bradley
5	100K 1W 5%		Allen Bradley
5	330K 1W 5%		Allen Bradley
5	1 meg 1W 5%		Allen Bradley
2	100 ohm 2W 5%		Allen Bradley
5	1K 2W 5%		Allen Bradley
5	10K 2W 5%		Allen Bradley
5	100K 2W 5%		Allen Bradley
5	47 ohm 1/2W 5%		Allen Bradley

<u>Quantity</u>	<u>Description</u>	<u>Part Number</u>	<u>Manufacturer</u>
Fuses			
5	3/10 amp	313.300	Littelfuse
10	1/2 amp slo blo	313.500	Littelfuse
10	1 amp slo blo	313001	Littelfuse
10	2 amp slo blo	313002	Littelfuse
10	5 amp	3AG	Littelfuse
5	10 amp ceramic	ABC	Buss
5	15 amp	AGC	Littelfuse
5	20 amp	3AG	Littelfuse
5	25 amp	3AG	Littelfuse
5	30 amp	3AG	Littelfuse
10	50 amp 250v replacement links	LKN	Buss
Tools			
1	components carrying case	47-K per NCR Specs.	Davis Sample Case Factory, L. A.
1	6" screwdriver	R-5166	Xcelite
1	3" screwdriver	R-3163	Xcelite
1	"stubby" screwdriver	S-3161	Xcelite
1	6" longnose pliers	226-S	Utica
1	6" diagonal pliers	184	Snap-on
1	insulated diode pliers	S-547	NCR Fab.
1	6" open-end adjustable wrench	AC-16	Crestoloy
1	Miti-Mite trouble light (with magnetic base)	200	Tiny Titan
1	35 W soldering iron	360	Drake
1 set	hex and spline wrenches	560	Walsco
1 lb.	60-40 multicore solder		Ersin
1	soldering aid		Hytron
1	Thickness gauge	172-A	Starrett
1	brush - cleaning type	B-15	Ames Supply Catalogue
Miscellaneous			
1 bottle	carbon tetrachloride	(4 oz.)	
1	wax pencil - white	130-W	
1 piece	1/8"x2"x10" soft cotton felt (for tape unit)		
15 feet	# 18 black test lead wire	8899	Belden
15 feet	#18 red test lead wire	8899	Belden
12	crocodile clips	85	Mueller
6	red insulators	87	Mueller
6	black insulators	87	Mueller

Flip-Flop and Memory Output Proposition Check Points

- A_1 (TC2 B1 5) 1D14K 1M53K 1M12K 2A13K 2C17K 2E1K 2E47K 2D2K
2I29K 2K4K 2O25K 2Q44K 2U39K 2W50K 2BB32K 2CC28K
4W51K 4AA51K
- A_1' (TC2 C1 5) 1D16K 1I18K 1M9K 1A38K 2A16K 2D43K 2K46K 2U37K
2X45K 2X51K 2BB30K 2CC26K 4T46K 4X51K 4BB51K 4N19P
- A_2 (TC2 B4 5) 1A40K 1I7K 1M44K 1O50K 1B49K 2A14K 2A48K 2C27K
2D48P 2K7K 2O27K 2U29K 2U34K 2CC25K 2CC27K 2M48K
4Y51K 4CC51K 4P29K
- A_2' (TC2 C4 5) 1I4K 1C29K 2A17K 2E35K 2F1K 2G46K 2P32K 2U31K
2V32K 2BB29K 2BB31K 4S25K 4Z47K 4DD51K
- A_3 (TC2 B6 5) 2Q48K 2R44K 2W37K 2DD25K 2DD28K 4W43K 4AA42K
- A_3' (TC2 C6 5) 1A36K 2C20K 2J52K 2K37K 2S27K 2S48K 2Y36K 2CC29K
2DD31K 4S28K 4X43K 4BB42K
- A_4 (TC2 B8 5) 1M32K 1O44K 1B29K 1A42K 1B37K 2D20K 2G32K 2M32K
2N28K 2P27K 2T32K 2CC30K 2DD26K 4Y43K 4CC42K
- A_4' (TC2 C8 5) 2D23K 2O53K 2P50K 2V26K 2DD29K 2DD32K 4Z39K 4DD42K
4DD32K 4R48K
- A_5 (TC2 B11 5) 1M11K 2C11K 2C18K 2D26K 2G27K 2G39K 2K11K 2L30K
2V37K
- A_5' (TC2 B11 3) 1A34K 2C8K 2CC46K
- A_6 (TC2 B12 5) 1A45K 1B32K 1J16K 1M31K 1A46K 2D21K 2M33K 2O54K
- A_6' (TC2 B12 3) 1A30K 1A32K 2D24K 2M34K 2O51K 2CC38K
- A_7 (TC2 B13 5) 1G2K 1K21K 2G20K 2I47K 2K32K 2Q27K 2Z47K 2BB47K
4U38K 4P47P
- A_7' (TC2 B13 3) 1J24K 1K24K 2Z36K 2BB51K
- A_8 (TC2 A14 6) 1K23K 2C45K 2M47K 2O34K 2Q29K 2BB47K 4P46P
- A_8' (TC2 A14 11) 1J23K 1K20K 2F45K 2BB43K
- A_9 (TC2 A15 6) 1O10K 2R36K 2S54K
- A_9' (TC2 A15 11)
- A_{10} (TC2 B14 6) 1O12K 4T49K
- A_{10}' (TC2 B14 11) 3L21K 4R45K
- A_{11} (TC2 B15 6) 1O14K

A_{11}' (TC2 B15 11)
 A_{12} (TC2 C15 6) 1D1K 2BB37K
 A_{12}' (TC2 C15 11) 1D4K
C (TC1 A9 5) 1CC53K 1B27K 1C27K 1E27K 1F27K 1G27K 1H27K 1J27K
1L27K 1M27K 1E29K 1I29K 1R31K 2A54K 2E54K 2H54K
2J54K 2L54K 2N54K 2Q54K 2S54K 2T54K 2W54K 2X54K
2Y54K 2Z54K 2AA41K 2AA54K 2BB40K 2BB54K 2CC40K
2CC44K 2CC54K 2DD46K 2DD54K 3G50K 3L14K 3M14K
3M20K 3N14K 3DD52K 4051K 4P54K 4V54K 4V50K 4V47K
4W38K 4W29K 4X38K 4E51K 4Y38K 4Z34K 4AA54K 4AA36K
4AA45K 4BB54K 4BB45K 4BB34K 4CC45K 4CC54K 4DD45K
4B53K
 D_1 (TC4 C8 3) 2L9K 4Y33K 4K2K
 D_1' (TC4 C8 5) 2L7K 4W32K 4K1K
 D_2 (TC4 D8 3) 2L8K 2L10K 4X33K
 D_2' (TC4 D8 5) 2L1K 2L2K 4X29K
 D_3 (TC4 D9 3) 2L11K 2N11K 4W33K
 D_3' (TC4 D9 5) 2M11K 2O11K 4X32K 4X25P
 D_4 (TC4 D10 3) 2L12K 2M12K
 D_4' (TC4 D10 5) 2O12K 2N12K 2X26P 4Y29K
 E_0 (TC2 B2 6) 4E54K
E (TC4 B1 5) 4A54K
 E' (TC4 B1 3) 4A52K
 E_1 (TC4 D1 5) 2C10K 2I4K 4B54K
 E_1' (TC4 D1 3) 4A50K
 E_2 (TC4 A2 6) 4C54K 2J24K
 E_2' (TC4 A2 11) 4B52K
 E_3 (TC4 B2 6) 4C50K
 E_3' (TC4 B2 11) 4C52K
 E_4 (TC4 C2 6) 2C52K 2E3K 2E17K 2H23K 4D52K
 E_4' (TC4 C2 11) 2F41K 4D54K
 E_5 (TC4 B3 5) 1N53K 1N44K 1N16K 1O38K 2A8K 2A8P 2A34K 2A38K
2A42K 2A51K 2B2K 2C7K 2C42K 2D6K 2D12K 2D27K
2D30K 2D49K 2E48K 2G4K 2G26K 2H5K 2H14K 2H52K
2I48K 2J2K 2J10K 2L46K 2M35K 2M46K 2Q43K 2Q51K
2R37K 2T38K 2U40K 2V50K 3O15K 4O9K 4S37K 4T43K
4N20P 4V34K

E_5' (TC4 C3 5) 2B4K 2D15K 2D47P 2D52K 2E38K 2F37K 2F48K 2F48K
 2G1K 2G36K 2G40K 2J48K 2K33K 2L31K 2O35K 2O50K
 2S44K 2S51K 2T41K 2V53K 2O21K 4Q33K 4S41K
 F_0 (TC1 A5 5)
 F (TC1 D1 5) 1A28K
 F' (TC1 D1 3) 1A25K
 F_1 (TC1 D2 6) 1B28K
 F_1' (TC1 D2 11) 1B25K
 F_2 (TC1 C2 5) 1A20K 1A8K 1A2K 1B2K 1C6K 1D5K 1I8K 1J12K 2B1K
 2D44K 2D53K 2I33K 2I51K 2J7K 2W42K 2CC47K 4N12K
 4N1K 4Q44K 4S53K 4T40K 4T30K 4V33K 4V38K
 F_2' (TC1 C2 3) 1A23K 1A5K 1I5K 2D5K 2D50K 2I16K 2J42K 2K28K 2AA43K
 G_0 (TC1 A5 6)
 G (TC1 A2 5) 1D28K
 G' (TC1 A2 3) 1D25K
 G_1 (TC1 B3 5) 1F19K 1E28K
 G_1' (TC1 B3 3) 1E25K
 G_2 (TC1 A4 6) 1J3K 1F28K
 G_2' (TC1 A4 11) 1F25K
 G_3 (TC1 B4 6) 1E30K
 G_3' (TC1 B4 11) 1F30K
 G_4 (TC1 C4 6) 1D17K 1F11K 1I11K 1G28K 2I34K
 G_4' (TC1 C4 11) 1I2K 1G25K 2K27K
 G_5 (TC1 D5 5) 1F2K 2G22K 1H12K 1H3K 1O21K 1E4K 2B20K 2B37K
 2C19K 2C41K 2F22K 2F49K 2I15K 2I18K 2L52K 2M29K
 2Y42K 4O5K 4P43K 4Q47K 4Q43K 4T34K 4T26K 4U34K
 G_5' (TC1 D5 3) 1H9K 2E27K 2N49K 2O29K 2O37K 2Z39K 2C22K 4P40K
 H_0 (TC1 A6 5)
 H (TC1 D6 5) 1I28K
 H' (TC1 D6 3) 1I25K
 H_1 (TC1 D7 6) 1J28K
 H_1' (TC1 D7 11) 1J25K
 H_2 (TC1 B7 5) 1C22K 1E20K 1H22K 1L19K 1L8K 1M8K 1M5K 1N19K
 1B40K 1D32K 1L49K 1AA33K 1DD42K 1I21K 4N49K 4P39K
 4Q29K 4V26K 4AA38K 4AA32K 4CC47K 4CC38K 4AA47K
 H_2' (TC2 B7 3) 1C19K 2N51K 4M16K 4N52K 4N16K 4N5K 4O19K 4P42K
 4S32K 4BB47K 4BB38K 4BB32K 4DD47K 4DD38K

J₀ (TC1 B10 6) 1K28K
J (TC1 A8 5) 1K25K
J' (TC1 A8 3) 1L28K
J₁ (TC1 B8 6) 1L25K
J₁' (TC1 B8 11) 1B19K 1D2K 1N50K 1N47K 1O41K 2F18K 2K1K 2T37K
J₂ (TC1 B8 5) 2V54K 3O18K
J₂' (TC1 D8 3) 2T40K 2V51K 2N21K 4Y27K
K (TC4 B10 3) 1A4K 1C18K 1H11K 2A10K 2D13K 2V38K 2Y47K 2B8K
K' (TC4 B10 5) 3L49K 4O16K
L₁ (TC4 B13 3) 1A7K 1C21K 1H8K 1CC52K 2A3K 2A23K 2D16K 2T28K
L₁' (TC4 B13 5) 2DD39K 3L15K 3M21K 3DD44K 4P33K 4R54K
L₂ (TC4 A15 3) 2C30K 2H17K 2R25K 3O13K
L₂' (TC4 A15 5) 2G18K 2G49K 3N13K 4P27K
L₃ (TC4 D13 3) 2I40K 2R27K 3O12K
L₃' (TC4 D13 5) 2K40K 3M12K
L₄ (TC4 C15 3) 2N32K 2R29K 3N9K
L₄' (TC4 C15 5) 2P47K 3O9K
L₅ (TC4 B15 3) 1K14K 2F24K 2R40K 3M10K
L₅' (TC4 B15 5) 2F20K 2S37K 3O10K 4M27K 4R36K 4R26K
M (TC3 C2 3) 1N54K 2F19K 2F23K 2R33K 4Y26K
M' (TC3 C2 5) 1N48K 2F16K 3L16K 3M22K 4M50K 4R39K 4R29K 4V40K
M_w (TC4 D6 5) 2F15K
M_w' (TC4 D6 4) 4V41K
M_s (TC4 D7 3) 4N50K
M_s' (TC4 D7 5) 4N53K
N₁ (TC3 D3 5) 4V43K
N₁' (TC3 B4 5) 4W26P
N₂ (TC3 C4 5) 3L28K 3M47K 4F10K 4F7K
N₂' (TC3 B5 5) 3L27K 3L47K
N₃ (TC3 C5 5) 3M29K 3O47K
N₃' (TC3 C6 5) 3M27K 3N47K
N₄ (TC3 C5 5) 3E21K 3E27K 3H31K 3I35K 3N31K 3Q47K 4A29K 4C39K
N₄' (TC3 C6 5) 3H34K 3H37K 3M43K 3N27K 3P47K 4B41K 4C42K 4D18K
N₅ (TC3 C6 5) 4L8K

N_4 (TC3 B7 5) 4B42K 4B15K 4C37K 4K23K 4B14K
 N_4' (TC3 B8 5) 3H33K 3H36K 3O27K 3R47K 3I32K 4A30K 4D41K
 4D19K 4F11K 4F8K 4K21K 4L9K 4D43K
 N_5 (TC3 C8 5) 3J45K 3U47K 4K49K
 N_5' (TC3 C9 5) 3K45K 3T47K 4L49K
 N_6 (TC3 B10 5) 3I46K 3W47K 4J48K
 N_6' (TC3 B11 5) 3K46K 3V47K 4L48K
 N_7 (TC3 C11 5) 3G47K 3Y47K 4H47K
 N_7' (TC3 C12 5) 3E24K 3K47K 3X47K 4L47K
 N_8 (TC3 B12 5) 3E23K 3D48K 3AA47K 4D46K
 N_8' (TC3 C14 5) 3K48K 3Z47K 4L46K
 N_9 (TC3 B15 5) 3CC47K 4L45K
 N_9' (TC3 B14 5) 3E22K 3K49K 3BB47K

Drawing List

MODEL 102-A COMPUTER

Computer - Binary General Purpose	20200001
Computer - Assembly	20217001
Flow Diagram	20243003
Logical Equations	20213297
Block Descriptions for Flow Diagram	20213296
Drawing Package	20240315

DRAWING PACKAGE

(Schematic and Reference Drawings)

Flow Diagram	20243003
Logical Equations	20213297
Block Description Flow Diagram	20213296
<u>Logical Networks</u>	
E_o Recirculating Register - Record Prop.	20213006
$F_o G_o H_o$ Recirculating Registers - Record Proposition	20213007
J_o Recirculating Register - Record Prop.	20213008
K Flip-Flop Grid Equations	20213009
l_1 to l_5 Channel Selector Grid Circuits	20213010
Flexowriter Conversion Circuits	20213011
a_1 to a_4 Grid Equations	20213012
Program Counter Sums	20213013
Thyratron Grid Equations	20213014
Program Counter Grid Equations	20213015
Program Counter Products	20213016
Pulse and Octal Digit Counter	20213017
a_5 to a_{12} Grid Circuits	20213018
Memory - Logical Circuits	20213383
<u>Schematics</u>	
Power Supply	20223001
Test Panel	20213082
Upper Tube Chassis #1	20223023
Upper Tube Chassis #2	20213104
Stabiline	20223377
Diode Board #1	20213027
Diode Board #2	20213030
Diode Board #3	20213033
Diode Board #4	20213036

Schematics (Cont'd)

Plug-in Chassis #1	20213262
Plug-in Chassis #2	20213263
Plug-in Tube Chassis #3	20213264
Plug-in Tube Chassis #4	20213265
Console	20253007

Plug-in Schematics

12BH7 Flip-Flop, Clamped	20263001
5687 Record	20263003
6BQ7A Pre-Amp	20263005
6BQ7A Pre-Amp Special	20263366
5687 Logical Driver	00263002
5687 Flip-Flop Driver	00363002
5963 Logical Driver	00463002
5963 Flip-Flop Driver	00563002
5881 Flip-Flop Driver	00663002
5687 Dual Phase Inverter	10663006
6BQ7A Clipper	10663007
12AT7 Schmitt Trigger	10663008
Dual 5687 Main Memory Limiter Amplifier	20263337
5687 One Shot, Clamped	10663028
Dual 6BQ7A Logical Relay Driver	12663101
Record Cathode Follower	20263329
12AT7 Flip-Flop, Clamped	20263011
12AT7 Flip-Flop, Unclamped	20263178
5681 Cathode Follower Driver	20263292
Dual 12AT7 Cathode Follower	20263009
Quadruple Flip-Flop Input Coupler	20263013
Dual 6AN5 Special Record Tube Driver	20263364
5963 Delay Line and Cathode	12763156
12AT7 One Shot, Unclamped	10563001

Component Installations

Memory Plug-in Units	20236279
Plug-in Tube Chassis #1	20216298
Plug-in Tube Chassis #2	20216299
Plug-in Tube Chassis #3	20216300
Plug-in Tube Chassis #4	20216301
Upper Tube Chassis #2	20216271
Wiring Diagram - Computer Intercomponent	20213002
Flexowriter Wiring Modification and Associated Circuits	20253003
Commercial Controls Corp. - Flexowriter Service Manual	-----
Arithmetic and Control Center Continuity Check	20213295
Arithmetic and Control Center Interboard Continuity Check	20213355
Arithmetic and Control Center Negative Check	20213352
Resistor Check - Diode Boards	20213359

POWER SUPPLY

Schematic	20223001
Wiring Installation - Misc.	20226238
Upper Power Supply No. 3 (UPS-3) - 225V Reg.	
Wiring Installation	20223005
Component Installation	20226188
Strip No. 1 - Component	20227173
Strip No. 2 - Component	20227172
Strip No. 8 - Component	20227277
Strip No. 9 - Component	20227278
Cable Assembly - UPS-3 Plug	20227163
Upper Power Supply No. 4 (UPS-4) - -300V Reg	
Wiring Installation	20223009
Component Installation	20226190
Strip No. 4 - Component	20227164

Cable Assembly - UPS-4 Plug	20227162
Lower Power Supply No. 1 (LPS-1) - Filament Chassis	
Wiring Installation	20223017
Strip No. 5 - Component	20227181
Lower Power Supply No. 2 (LPS-2) - Stabiline Chassis	
Schematic - Stabiline	20223377
Wiring Modification - Stabiline	20226376
Lower Power Supply No. 3 (LPS-3) - 225V Unreg.	
Wiring Installation	20223021
Lower Power Supply No. 4 (LPS-4) - Relay Chassis	
Wiring Installation	20227010
Strip No. 3 - Component	20227174
Strip No. 7 - Component	20227276
Cable Assembly - LPS #4 Plug	20227161

MISCELLANEOUS CHASSIS

Upper Tube Chassis No. 1 (UTC-1) - IBM Thyatron Chassis	
Schematic	20223023
Wiring Installation	20213087
Thyatron Strip - Wiring Install.	20213126
Thyatron Strip - Wiring Install.	20213127
Component Installation	20216286
Cable Assembly	
Input Plug, UTC-1	20213110
Output Plug, UTC-1	20213112
Upper Tube Chassis No. 2 (UTC-2) - Fill and Flexowriter Relays	
Schematic	20213104
Wiring Installation	20213088
Component Installation	20216271

Plug-in Units

Assembly No.	Schematic No.	Req. No.
00467001	00463002	4
12667102	12663101	5
20267396	20263395	1

Cable Assembly

Input Plug, UTC-2	20213111
Output Plug, UTC-2	20213113

Fan Contactor Panel

Wiring Installation	20226222
---------------------	----------

Test Panel

Schematic	20213082
-----------	----------

Wiring Installation	20213156
---------------------	----------

Cable Assembly	20217375
----------------	----------

Cable Connector Panel

Cable Assembly

Console Receptacle	20213114
IBM Reader Receptacle	20253009
IBM Punch Receptacle	20253008
Tape Receptacle #1	20253002
Tape Receptacle #2	20253006
Power Receptacle	20257259
Power Cable	20200006

Contactor Panel

Wiring Installation	20226177
---------------------	----------

Fuse Panel

Wiring Installation	20226189
---------------------	----------

Component Installation	20226192
------------------------	----------

Installation - Arithmetic and Control Center	20216312
--	----------

Wiring Installation - Interboard Terminal Board	20213005
---	----------

Wiring Installation - Inter-Chassis Terminal Strips	20213115
---	----------

DRAWING LIST

I-7

<u>Diode Board</u>	#1	#2	#3	#4
Schematic	20213027	20213030	20213033	20213036
Wiring Installation				
External	20213237	20213236	20213235	20213234
Resistor	20216321	20216324	20216323	20216322
Resistor Strip	20216106			
Miscellaneous	20216328	20216327	20216326	20216325
Horizontal Strapping	20213063	20213068	20213073	20213078
<u>Tube Chassis</u>	#1	#2	#3	#4
Schematic	20213262	20213263	20213264	20213265
Wiring Installation	20213038	20213042	20213044	20213046
Component Installation	20216298	20216299	20216300	20216301

Plug-in Units

<u>Assembly No.</u>	<u>Schematic No.</u>	<u>No. Req.</u>	<u>No. Req.</u>	<u>No. Req.</u>	<u>No. Req.</u>
00267001	00263002	2	10	2	2
00367001	00363002	4	3	1	9
00467001	00463002	21	12	3	3
00567001	00563002	-	-	1	6
00667001	00663002	1	8	34	2
10567002	10563001	1	-	-	-
10667016	10663006	2	-	1	2
10667017	10663007	4	-	1	2
10667018	10663008	1	-	-	-
10667029	10663028	1	-	-	2
12667102	12663101	1	3	-	-
12767157	12763156	1	-	-	-
20267010	20263009	4	9	4	9
20267012	20263011	8	6	1	3
20267014	20263013	-	-	-	2
20267179	20263178	4	7	12	15
20267293	20263292	1	-	-	-
20267365	20263364	2	-	-	-

MEMORY CHASSIS

Wiring Installation

Memory Chassis	20233011
----------------	----------

Memory Chassis to Diode Boards	20233012
--------------------------------	----------

Component Installation	20236279
------------------------	----------

Plug-in Units

<u>Assembly No.</u>	<u>Schematic No.</u>	<u>No. Required</u>
20267002	20263001	5
20267004	20263003	3
20267006	20263005	8
20267339	20263329	8
20267340	20263337	1
20267367	20263366	2

Installation

Memory	20236253
--------	----------

Head to Memory Chassis Connections	20233013
------------------------------------	----------

Assembly	20237015
----------	----------

Magnetic Head and Mount Assembly	14137001
----------------------------------	----------

Magnetic Head and Mount Assembly	14137002
----------------------------------	----------

Magnetic Head and Mount Assembly	14137003
----------------------------------	----------

Magnetic Head and Mount Assembly	14237003
----------------------------------	----------

Magnetic Head and Mount Assembly	14137006
----------------------------------	----------

Magnetic Head and Mount Assembly	14237009
----------------------------------	----------

Magnetic Head and Mount Assembly	14237010
----------------------------------	----------

Magnetic Head and Mount Assembly	14137007
----------------------------------	----------

CONTROL CONSOLE AND FLEXOWRITER

Control Console

Assembly	20257001
----------	----------

Schematic	20253007
-----------	----------

Cable Assembly	
----------------	--

DRAWING LIST

I-9

Console	20256221
Console to Computer	20200002
<u>Flexowriter</u>	
Wiring Modification Diagram	20253003
Component Installation	20256368
IBM	
IBM Modifications (for use with 102-A system)	20253010
Modifications for Punching IBM	20253021